# Feature-Based Hierarchical Knowledge Engineering for Aircraft Life Cycle Design Decision Support

A Dissertation
Presented to
The Academic Faculty

By

**Wei Zhao**

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy

School of Aerospace Engineering
Georgia Institute of Technology
May 2007

# Feature-Based Hierarchical Knowledge Engineering for Aircraft Life Cycle Design Decision Support

Approved by:

Prof. Dimitri N. Mavris, Advisor
Aerospace Engineering
Georgia Institute of Technology

Prof. Jye-Chyi Lu
Industrial System Engineering
Georgia Institute of Technology

Prof. Vitali Volovoi
Aerospace Engineering
Georgia Institute of Technology

Prof. Daniel P. Schrage
Aerospace Engineering
Georgia Institute of Technology

Dr. Zhimin Liu
Aerospace Engineering
Georgia Institute of Technology

Date Approved: April 9, 2007

*To my wife, Yufeng Yao*

*To my parents, Fuyang Zhao and Guqing Fang*

*To my lovely daughters, Danielle Zhao and Annie Zhao*

# ACKNOWLEDGEMENTS

Reaching this point would not have been possible without the help of others. I would like to thank my advisor, Dr. Dimitri Mavris. I can not express how much I appreciate his support over the last four years. He provided a perfect balance of freedom and guidance during the course of my study.

I would also like to express my deep gratitude to Dr. Jye-Chyi Lu for guiding me in the world of knowledge mining, for his willingness to share his ideas in research problems, and for the energy that he put into advising me on my dissertation work.

Special thanks go to Dr. Liu for his valuable comments and ideas, continued encouragement, and support throughout my research. Dr. Liu has been a close friend.

I would also like to thank Dr. Schrage and Dr. Volovoi for their valuable feedback on this dissertation.

Reaching this point in my life would not have been possible without the unconditional love of my parents. I thank them for always being there for me: Mr. Fuyang Zhao and Mrs. Guqing Fang.

I especially treasure the support, inspiration, and patience from the love of my life, Yufeng Yao. She is a blessing to me.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS

| | |
|---|---|
| $\beta$ | Shape parameter in Weibull distribution |
| $\gamma$ | Location parameter in Weibull distribution |
| $\eta$ | Scale parameter in Weibull distribution |
| $\mu$ | Mean of a variable |
| $\sigma$ | Standard deviation of a variable |
| $\Gamma(n)$ | Gamma function |
| $f(T)$ | Probability density function |
| $R(T)$ | Reliability function of Weibull function |
| $R_i$ | Output model for response variable in neural network models |
| $H_j$ | Hidden function built from input variable in neural network models |
| *Squish* | The S-shaped activation function in neural network models |

# SUMMARY

The design process of aerospace systems is becoming more and more complex. As the process is progressively becoming enterprise-wide, it involves multiple vendors and encompasses the entire life cycle of the system, as well as a system-of-systems perspective. The amount of data and information generated under this paradigm has increased exponentially creating a difficult situation as it pertains to data storage, management, and retrieval. Furthermore, the data themselves are not suitable or adequate for use in most cases and must be translated into knowledge with a proper level of abstraction. Adding to the problem is the fact that the knowledge discovery process needed to support the growth of data in aerospace systems design has not been developed to the appropriate level. In fact, important design decisions are often made without sufficient understanding of their overall impact on the aircraft's life, because the data have not been efficiently converted and interpreted in time to support design.

In order to make the design process adapt to the life cycle centric requirement, this thesis proposes a methodology to provide the necessary supporting knowledge for better design decision making. The primary contribution is the establishment of a knowledge engineering framework for design decision support to effectively discover knowledge from the existing data, and efficiently manage and present the knowledge throughout all phases of the aircraft life cycle. The second contribution is the proposed methodology on the feature generation and exploration, which is used to improve the process of knowledge discovery process significantly. In addition, the proposed work demonstrates several multimedia-based approaches on knowledge presentation.

# CHAPTER I
# INTRODUCTION

## 1.1 Motivation

The aerospace industry is continuously evolving to meet customer demands since the first airplane flew 100 years ago. All the areas in this industry, such as the design disciplines, production processes and airline operations, have established a set of methods to accomplish the needed tasks within their perspective domains. However, as a whole, the overall life cycle considerations are still hindered by the limitations and the relative isolation of available knowledge across all its components. The people and tools are configured to work in narrow fields with little or no consideration across those fields.

For example, the aircraft designers normally design an aircraft for the best performance, such as speed, range, weight, and specific fuel consumption, and they may consider certain manufacturing issues if concurrent engineering is used to involve production engineers. However, the operation and support concerns, such as the airport locations, flight frequencies, take-off and touchdown cycles, ground maintenances, are seldom brought into the picture. It is ironic that the issues present in the operation and support phases are not addressed enough in the design phase, although they represent most of the life of an aircraft and are perhaps the most important phases for the success of an aircraft development and sale.

If knowledge from all phases is synthesized in early decision-making and becomes available to the decision makers, then aircraft can be made more suitable and practical to the end users, and will consequently present significant efficiency and cost savings. Unfortunately, in the early design phases of an aircraft life cycle, tasks are locally optimized within their own scopes, and ignore or largely simplify the effect of other consequential phases. It is perhaps best described by an old Chinese story called the blind men touching an elephant as displayed in Figure 1.1.1

The story goes as follows. Several blind people ran into an elephant, and they are trying to figure out what it looks like. One person touched one leg of the elephant, and claimed it is like a big moving pillar; another person climbed to the side of the elephant, and argued it is like a tall wall; the third person held the trunk of the elephant, and cried that he found a huge snake...

Figure 1.1.1 Blind men touching an elephant

Because none of the blind men saw or felt through touching the whole creature, none of them was able to identify the elephant correctly. Without considering the overall aircraft life cycle, in a holistic manner, the decision makers could reach erroneous conclusions. Furthermore, it could cause huge inefficiencies if the following phases are not considered together early on, since it is usually difficult to modify established designs and procedures later on. Although it is ideal and it does make sense to consider everything a priori, several realistic factors prevent this from happening.

1) Aircraft fall under the category of complex systems. The engineers already have their hands full with what needs to be done to design and certify a given vehicle, so they rarely have extra resources, time, and funds to handle additional information. In some cases they even lack the proper means to process or account any additional considerations. Furthermore, the aircraft design is migrating towards a more system-of-systems perspective, which posts an even greater challenge on current design solutions.

2) The subsystems are isolated from each other, e.g. different vendors and suppliers, thus knowledge is kept within the individual sectors, and it is not usually shared across these boundaries. There are various obstacles on knowledge sharing, such as inter-industry barriers, intra-industry competitions, company policies, and human factors.

The term "knowledge" above has various definitions depending on the topic or perspective. So to avoid any confusion with the use of some of the terms used throughout this thesis, the following table provides the definitions used by the author.

Table 1.1.1 Terminology Definitions

| | Definition in this thesis | Example |
|---|---|---|
| Data | Facts or pieces of information | One maintenance log entry for Delta aircraft 7008 on 8/22/2003 |
| Information | Organized data that can be communicated | A set of maintenance records on Delta's Boeing 777 fleet between 4/2003 and 12/2003 |
| Knowledge | Understandings or ideas drawn from information, which can be described explicitly for a specific purpose | The In-Flight Entertainment (IFE) system on Delta's Boeing 777 is unreliable in general |
| Wisdom | Selection of appropriate knowledge to make decision for a specific task | Knowing how to improve the IFE reliability on Delta's Boeing 777 |

Information is organized data, and knowledge is obtained from information, wisdom is the proper selection of knowledge. The focus in this thesis is to establish a systematic process to obtain knowledge from data and information, and effectively manage it. The creation of wisdom is a separate topic, and not in the scope of this thesis.

3) There is a lack of an organized, structured knowledge management and presentation framework. People do not have a centralized resource repository where they can find the related knowledge, or they are not even aware of the existence of such knowledge.

4) Such knowledge is unavailable. Even if the other sectors wish to share, they may not have enough knowledge on hand. Although it is becoming easier and easier to gather data, and more data are accumulated with the advantage of information technology, it may not necessarily result in sufficient knowledge. Knowledge could still be missing due to the lack of ability to understand, organize and utilize the information.

The above issues are like the sound barrier to a World War II aircraft designers, but it's not impossible to overcome, and the results will be promising if one overcomes such barriers. A possible solution is concurrently considering and accounting for the various life cycle phases as early in the design phase as possible and by abiding to a well structured systems engineering approach. Many advances have been made in the system engineering field to facilitate and improve the design process [Mejzak 1991, Gaffney 1994, Happel 1998, LaBerge 1999, Brewer 2003], but they are usually limited to certain subsystems, such as communication system, navigation system, and so on. [Marx 1998] proposes an integrated design and manufacturing approach allowing economic decision based on holistic system design with a special focus on integrated cost and engineering models. [Baltes 2002] presents a test support system for the different phases of the aircraft life cycle. However, the following issues are not addressed thoroughly. How does one obtain knowledge from all phases of the aircraft life cycle? How does one manage existing knowledge efficiently? How does one establish methods to present the knowledge effectively to facilitate design decision making.

In this thesis, the focus is on knowledge generation, structured management, and presentation of knowledge over the aircraft life cycle to facilitate system design. It is essential to establish a process to discover knowledge across the aircraft life cycle from existing information, and present it to directly support design decision making. The development of such a process is the ultimate goal of the proposed research. Moreover, the proposed process is not limited to the design phase, and it can be used to support other phases of the aircraft life cycle.

### 1.1.1 Complexity of the System

Aircraft system design, development and operation are complex processes. The associated activities are not usually handled by an individual company or organization. From the concept creation to the retirement of one type of aircraft, it takes several decades. Iterations within such a long-lived product life cycle are always required to consolidate baselines in design, development, production, operation, and support.

In general, major life cycle activities of an aircraft system can be classified into the following phases, in Figure 1.1.2. It is presented in a waterfall model.

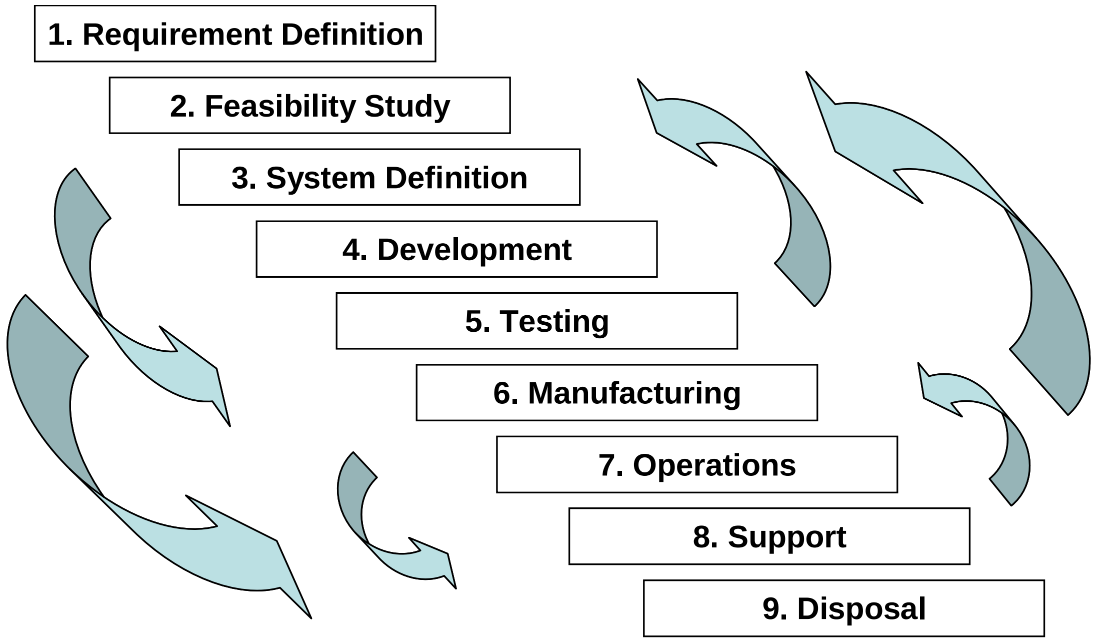



Figure 1.1.2 Phases of Aircraft Life cycle Activities

In the aerospace industry, due to the complexity of the system, the above phases are supported by specific engineering groups and disciplines. The engineering activities traditionally aim to consolidate the system baseline and its implementation incrementally. This process is carefully controlled by adopting specific configuration management and product assurance methodologies, which increased the complex of the aircraft system.

The increasing complexity of the aircraft system process often results in large interdependent task groups due to the nature of the concurrent strategy. The larger the size of interdependent task groups is, the more difficult the team organization is, and thus the more chance of delays in the system process. Moreover, as complexity increases, it becomes more difficult to manage the interactions among tasks and people; it may be impossible to predict the impact of a single design decision throughout the development process [Zu 2004]. Therefore, it’s crucial to improve the information sharing mechanism in order to shorten the development time and lower life cycle costs.

### 1.1.2 Isolation of Systems in the System

Although all subsystems are supposed to form a complete system, knowledge is normally kept within the individual subsystems. There are various obstacles on knowledge sharing, such as inter-industry barriers, intra-industry competitions, company policies, and human factors.

Inter-industry barriers: due to the heterogeneity between systems, the data produced by the systems at one company cannot be read by people from another company. "One barrier to collaboration is the lack of interoperability among the application systems of different companies." [Hardwick 1997]

Intra-industry barriers: the nature of the integration requires cooperation, collaboration, mutual understanding, and sharing of resources. The traditional culture was seen to be hostile to this integration, which created a serious barrier in many institutions. Some typical issues are competition and territorialism, resistance to change and to new technologies, and incentives [Currier 2002]. Interpersonal trust is also an issue when people are facing a geographically distributed work environment [Zolin 2004].

Other system barriers in collaboration are in the form of perception, change, intentions, and temporal [Reddy 2004]. On the other hand, even if the barriers can be overcome, the amount of information resulting from knowledge sharing is overwhelming.

### 1.1.3 Lack of Structured Knowledge Management

In recent decades, our capacity of both generating and collecting data has been increasing expeditiously. With the rapid information growth, people have the chance to gain more and more knowledge out of it. However, there is a problem that always exists:

the management of the existing knowledge. It is hard to locate, or even know the existence of, the place where the related knowledge can be found. International Data Corp., a technology focused research group, estimates that poorly managed knowledge costs Fortune 500 companies about $12 billion per year [Stewart 2002]. The reasons for the lost money are: "substandard performance, intellectual rework, and a lack of available knowledge management resources."

If companies want to build, support, and optimize their business activities, knowledge is important. The right processes are required to capture and share knowledge across organization boundaries, which result in increased process efficiencies, maximized strategic effectiveness and improved innovative capabilities.

### 1.1.4 Need Efficient Knowledge Conversion from Rich Data

The fast-growing infrastructure of the information technology has enabled significant change in global economy. With the help of computers, internet, and automated data procurement processes, companies and organizations can easily gather large amount of information from multiple channels. The rapid growth of information resource has not been paralleled with the capability of knowledge extraction. As a result, we have mountains of data stored, but the knowledge we obtained so far is not sufficient, or fully utilized. Data collected in large databases become data archives that are rarely visited. Consequently, important decisions are often made not based on the information-rich data stored in databases, but rather on the decision maker’s intuition, simply because the decision maker does not have the tools to extract the valuable knowledge embedded in the large amounts of data. In addition, current expert system technologies typically rely on users or domain experts to input knowledge into knowledge bases manually. This procedure is prone to bias and errors, and it is extremely costly and time-consuming. The gap between data and knowledge is widening.

Noticed the importance of extracting knowledge from data, NASA is actively working on developing and applying new approaches on data analysis. A NASA/NIA workshop was held in 2004 to discuss perspectives in trending issues and strategies. Schaible pointed out "NASA lacks 'Value added' independent assessment of technical issues within its program and institutions" [Schaible 2004]. Otero also indicated that the current tools of data and trending analysis at NASA generally lack sophistication and automation, and inhibit decision support. Extensive 'hands-on' examination and analysis is needed to process data into meaningful information [Otero 2004].

### 1.1.5 Need Capable Methodology to Bridge the Gap, and Support Decision Making

The learning capability has to be expedited if one is to master the increasing complexities of today's aircraft life cycle. Best practice in knowledge engineering can be introduced to increase the chances of success.

Using a hierarchical format to organize and present knowledge, knowledge maps can be developed to make knowledge easy to find. One who succeeds in leveraging its knowledge capital is more likely to improve process efficiency and create new value in the organization. The competitive advantage in the aircraft life cycle can be enhanced or enabled by discovering, handling and presenting knowledge effectively. Effective knowledge management also improves product and process innovation. Companies will spend fewer resources reinventing the wheel through effective knowledge management, which will in turn enable their people to devote their time and talents towards achieving the organization's corporate goals.

Our goal is to develop a methodology that will bridge the gap between data, knowledge and decision making. The upfront and holistic attention towards aircraft life cycle considerations will lead to faster, more affordable design, production or manufacturing, operation and support of high-quality systems that satisfy all customer requirements, and minimizes the rework needed to comply with them after the fact. In the end, this approach may be viewed as a risk mitigation strategy.

## 1.2 Thesis Organization

The issues related to knowledge engineering for aircraft life cycle design decision support and the need for this research has been outlined in this chapter. Chapter II captures previous efforts in the literature relevant to the topic of this thesis. In Chapter III, research questions and hypotheses are presented, and key technical challenges are discussed with potential solution alternatives. Chapter IV describes the technical approach and methodology in this research. Implementations of the proposed methodology are illustrated in Chapter V. Finally, a brief recapitulation of the present research and proposed future work is addressed in Chapter VI.

# CHAPTER II
# BACKGROUND

## 2.1 Aircraft System Life cycle

The life cycle of a system in general has seven phases: (1) system requirements discovery, (2) alternatives evaluation, (3) full-scale engineering design, (4) implementation, (5) integration and system test, (6) operation, maintenance and evaluation and (7) retirement, disposal and replacement. However, the system life cycle is different for different industries, products and customers. [Chapman 1992, Wymore 1993; Kerzner 1995, Shishko 1995]

NASA (1995) divides the project life cycle into the following five phases. Although a project is not necessarily covering the whole life of a product, they have some level of similarity.

(1) Preliminary Analysis: in this phase, the feasibility and desirability of a suggested new major system and its compatibility with stated strategic plans are determined. It is to “make sure the project is worthwhile”.

(2) Definition: in this phase, the project is defined in enough detail to establish an initial baseline that is meeting mission needs. It is to "define the project and establish a preliminary design".

(3) Design: the detailed design of the system and its associated subsystems, including its operations systems are completed. It is to "complete the system design".

(4) Development: the subsystems, including the operations system, is built and integrated to create the system, meanwhile confidence on meeting the system

requirements is developed, and then the system is deployed and ensured to be ready for operations. It is to "build, integrate, and verify the system, and prepare for operations".

(5) Operations: the system is operated to meet the initially identified need or to grasp the opportunity, then to dispose of the system in a responsible manner. It is to "operate the system and dispose of it properly".

In the aircraft life cycle process, the system design and development baseline evolution is refined up to the point at which specific review events can take place to consolidate the defined baseline.

For example, in a typical aircraft system design and development program, subcontractors are often involved. Their roles can range from manufacturing of equipment/products to a larger responsibility encompassing the definition, design, and development of a complete subsystem, e.g. the propulsion subsystem. The relationships between a prime contractor, which is responsible of the overall program, and a subcontractor is regulated by contracts based on detailed definition of the work to be done, as well as of the products to be delivered.

From an industrial point of view, an aircraft system life cycle can be described in the following phases.

### 2.1.1 Aircraft Design

Aircraft manufacturers identify and answer the customer needs, then start to design an aircraft to meet specific requirements, such as capacity, speed, range, and so on. Some government regulations will also be involved, such as noise, pollution limits. It can be further divided into the following sub-phases, Conceptual design, where aircraft is treated as a whole, and many alternatives of concepts are created, evaluated and compared to better meet the customer requirements with consideration of cost and availability of technologies [Mavris 1995, Hines 2000, Hollingsworth 2000, Mavris 2002, Kamdar 2003]. The aircraft conceptual design can usually, to some extent, be regarded as variations of a baseline. Uncertainty can also be taken into account [Mavris 1997, DeLaurentis 1997, DeLaurentis 2000]

Preliminary design. After establishing the initial layout, the aircraft design is decomposed into disciplines, such as aerodynamics, structure, propulsion, and control [Rohl 1994, Mavris 1998, Mavris 1999, Kirby 1999], and then synthesis is performed to integrate them back together [DeLaurentis 1996]. Some Multi-disciplinary optimization is also involved [Mastroddi 2002, Piperni 2004, Bernardini 2004].

Detailed design. Configurations are further refined, and prototypes are tested to prepare for production.

### 2.1.2 Aircraft Manufacturing

The production of aircraft is commonly accomplished by certain decomposition and recomposition processes. Aircraft manufacturers order major components, such as engines, fuselage, and electronic systems, from component-manufacturing companies. Component manufacturing companies subcontract parts to part-manufacturing companies. Then when part manufacturing companies build these parts, major components are built based on these provided parts, and the aircraft is subsequently assembled by the aircraft manufacturers before it is delivered to customers, such as the military, airlines, individuals, etc. Quality, consistency and productivity are major drivers during the phases of the product life cycle.

### 2.1.3 Aircraft Operations

The operations of aircraft are mainly classified into three categories, i.e. airline business, military operations, and general aviation.

Air travel is the latest developed mode of transportation. After the beginning of the first commercial flight, air travel has been growing steadily, because of the speed and convenience provided by this mode. The farthest points on earth are reachable in a matter of hours by air transportation. Air carriers are either private or commercial. Commercial air carriers are divided according to their sizes or the type of services they provide. Size is determined based on the annual revenues. Three types of airlines are recognized: major airlines, such as American Airlines and Delta Airlines; national airlines, such as Southwest Airlines and Airtran airlines; and regional airlines, such as American Eagle Airlines and Arizona Express Airlines. Different types of service are offered: Cargo only; air taxi, which offers passenger service on demand; fractional ownership air transportation, which serves the customer, who has a portion of the ownership of an aircraft, with advanced notice; commuter, which offers passenger service based on published timetables; charter, for which the route and schedule are negotiated in a contract.

The advantages of the air transport are fast terminal-to-terminal transportation, reliable service (except under severe weather conditions), and attention to the customer through in-flight services and entertainment. Limited frequency of flights, capacity restrictions, and poor service of small cities are disadvantages. Also, long travel times to and from airports, which are typically located at the outskirts of urban areas, as well as long wait times, in check-in, security check, boarding, taxiing, baggage claim, increase the overall travel time. To overcome the above weakness of the traditional commercial air transport, several alternatives are promising: on-demand air transport, such as fractional ownership program, is getting more and more attention to business travels due to its flexibility and affordability; personal air vehicle (PAV) research is currently supported by NASA to create next generation aircraft for personal use.

For all types of airlines, the operational cost is always a major concern to the business. The cost of airline operation consists of the fixed costs and variable costs. Fixed costs include the aircraft fleet and maintenance facilities, computer reservation systems (CRS), management, logistics, airport counters, gates, and baggage-handling facilities, as well as offices in cities. Several of these may be leased, including aircraft, which makes them semi-variable in nature. Variable costs include labor and fuel (which combined account for 65% of the total variable costs [ATA]), landing fees (which cover the use of local, state or federal facilities), maintenance, etc.

As for airline operations, the airlines offer flights that carry passengers or cargo from various origins to various destinations. In general, the marketability of the service is determined by the timeliness, accuracy, functionality, quality, and price of the service [Yu 1998]. As perceived by air transportation customers, these criteria translate into flexible schedules, on time flights, safety, satisfactory in-flight services, proper baggage handling, and convenient ticket purchases. The competition in the air transportation market is fierce especially in the US, and the FAA regulations post tight restrictions on the operations as well. To meet the challenge and provide a service with high quality and low cost, airlines spend tremendous resources and effort to generate profitable and cost effective fare, flights schedules, fleet plans, aircraft routes, crew pairings, gate assignments, maintenance schedules, food service plans, training schedules, and baggage handling procedures. How to gain and keep a leading edge in the competitive air transportation market and how to make proper decision to run the airline effectively and efficiently to respond to customers' needs are the issues faced by top management of each airline.

Among all the problems faced by the air transportation system, airport congestion is becoming increasingly important. Airport congestion is observed when the number of arriving and departing aircraft reaches or exceeds the capacity of a field. At congested airports, arriving aircraft are placed on a holding pattern (usually spirals in the airspace near airport) and departing aircraft are queued on taxiways. Larger aircraft are part of the solution for this problem, since they require fewer landing and takeoff slots for serving a fixed number of passengers. Therefore, consolidation of flights and use of larger aircraft less frequently may offer some relief to congested airports by decreasing the required number of operations per passenger served. Although using larger aircraft has the economies of scale and may somewhat ease airport congestion, the concern of the low load factor may rise up if demand are not enough to fill the seats in larger aircraft.

### 2.1.4 Aircraft Support and Maintenance

The maintenance of aircraft is extremely important in any phase of operation, due to its effect on reliability, reputation, and the economics of the airline. Passengers generally cannot see the quality state of maintenance on aircraft engines, accessories, or structural parts. But they are immediately affected by any mechanical problem which causes schedule delays or which becomes visibly evident on the aircraft, as in the instance of liquid leaks or visible mechanical failures, such as engine failures, tire blowouts, heater failures, or door malfunctioning.

All maintenance irregularities and difficulties can be convincingly attributed to poor engineering design, faulty construction, aging, abnormal external conditions, improper operating techniques.

In general, maintenance consists of line maintenance, overhaul, and repair, which are defined as follows:

Line Maintenance: all mechanical work performed on aircraft, engines, and accessories without completely removing them from service. It includes all maintenance accomplished during the time between airplane’s revenue flights. In some cases, an active airplane may be held in line maintenance for 24 to 48 hours in addition to its normal flight schedule. It often includes replacement of units that are themselves subject to overhaul before reinstallation.

Overhaul: all mechanical work performed on the aircraft, engines and accessories which necessitates removal from service on a periodic basis and which results in the return to either a new condition or a condition considered by the airline equivalent to a new condition as service overhaul is concerned.

Repair: any unscheduled maintenance that must be accomplished because of unexpected damage to aircraft engines and accessories. The damage is defined as unexpected, although it may reoccur with such regularity as it maybe found in regular periodic inspections. Repairs may require the aircraft to be removed from service for many months, as in the instance of major aircraft accidents where the aircraft is still repairable.

Although airline operation utilizes the three classes of maintenance, it is difficult sometime to distinguish them completely from each other. In both line maintenance and overhaul, there are two systems of maintenance with different focuses:

Preventive maintenance: aircraft, engines, and accessories are subjected to a system of periodic replacement of parts regardless of their condition. For instance, engines will be completely removed after a specified length of service, no matter its apparent condition, and are overhauled. During the overhaul procedure, quite a few structural parts will be replaced by new parts regardless of their condition. Other parts will be replaced every second or third overhaul period. These periods of engine overhaul and parts replacement are based upon the experienced judgment of airline and manufacturer experts. The periods used must be approved by the FAA and they are gradually extended in length according to accumulated experience and improved materials.

Corrective maintenance: by this system aircraft, engines, and accessories are subjected to repair or replacement of parts when inspection determines the maintenance is required. For example, aircraft turbo-fan engine blades are given periodic examinations for nicks, bends and other damage. Unless some damage occurs from rocks on the run-up ramp, water on the runway, or some other source, nothing is done to the blades until they are finally removed with the entire engine for general overhaul. Whenever damage is discovered and need to be repaired, the fan blades are thereby subjected to corrective maintenance.

Line maintenance and overhaul utilize both preventive and corrective system of maintenance, and repair utilizes only the corrective system of maintenance.

Maintenance procedures: an airline must be sure every effort is made to accomplish each individual maintenance job with full satisfaction as to operational reliability and accomplishment in a minimum of man-hours in the shortest elapsed time. Most maintenance jobs can be accomplished by any of several procedures, where one is best with respect to reliability and efficiency. Airlines are putting effort to standardize whenever possible upon the best maintenance procedures. Such standardization substantially reduces the educational time necessary to introduce new aircraft and improves dependability and efficiency of aircraft maintenance.

Trouble shooting: one of the most fertile fields for improved maintenance procedures and equipment is the analysis and correction of mechanical irregularities wherein no obvious mechanical failures are involved. This general problem in trouble shooting has led to the development of special equipment and techniques often particular to each type of airplane. Individual experience has always been the strongest factor relied upon in aircraft-maintenance trouble shooting. The development of improved procedures and equipment will aid, but not replace, the experience factor. Replacement of a defective unit, instead of correction, is a practice of trouble shooting which has done much to minimize the time required for aircraft maintenance. The balance between the time required for changing, the time required for correction without changing, and the value of the inventory must be analyzed in evolving proper procedures in this aspect. The procedure selected must remain flexible because of variations in inventory as the maintenance requirements vary.

### 2.1.5 Aircraft Flight Safety

Aircraft flight safety attracts consistent attention in the entire aircraft life cycle, and continuous efforts took place from many aspects to enhance flight safety. Among them, Flight Operational Quality Assurance (FOQA) programs are safety programs designed to improve aviation safety through the proactive use of flight-recorded data. It involves the collection and analysis of data recorded during flight to improve the safety of flight operations, air traffic control procedures, and airport and aircraft design and maintenance. FOQA programs had started since 1995. Although FOQA is not required by FAA, airlines are encouraged to implement the program. As of March 2004, 13 airlines and about 1,400 airplanes were equipped with FAA-approved FOQA programs [PBI 2004].

The objective of FOQA is to use data recorded during flight to early detect technical flaws, unsafe practices, or unusual conditions, so that timely intervention is allowed to avert accidents or incidents. Airlines equip aircraft with specialized devices, called quick-access recorders (QARs), to continuously record hundreds of flight-data parameters from aircraft system and sensors. QARs capture flight data onto removable media, and then periodically data are delivered or downloaded to the ground analysis system at a centralized location for further processing, such as analyzing the data, identifying the trends, and taking actions to correct problems. For example, the data are evaluated against certain predefined events, such as the descent rate during approach, for deviations from the airline's specified tolerance thresholds. Deviations are flagged and evaluated by a monitoring team to determine their validity and understand possible causes, before the team proposes and evaluates corrective actions. Periodically, airlines aggregate the deviations over time to determine and monitor trends.

Although the program is primarily a safety program, airlines have reported financial benefits as well [GAO staff 1998]. With additional data on aircraft systems and engine, airlines are better equipped to achieve optimum fuel consumption and avoid unnecessary engine maintenance. Less incidents/accidents and lower insurance premiums result in lower cost over time. The primary difficulty to the implementation of FOQA programs are the data-protection issues. To resolve this issue, the FAA's FOQA protection rule, FAR 13.401, promises that the FAA will not use the operator's FOQA data for enforcement purposes except for criminal and deliberate acts. Under Part 193, the FAA promises to protect voluntarily disclosed information under an FAA approved FOQA program, from FOIA requests. The FAA promises to convey the need for the same protections to other federal agencies where it shares information.

FOQA data are also used by Federal Aviation Administration (FAA) for safety purposes [FAA 2004]. Through access to aggregate FOQA data, the FAA can identify and analyze national trends and target resources to reduce operational risks in the National Airspace System, Air Traffic Control, flight operations, and airport operations. The ground data replay and analysis system (GDRAS), which is a sophisticated software application, transforms flight-recorded data into a usable form, analyzes the data and detects events, and generates reports for review.

## 2.2 Collaborative Design

As industries and governments around the world restructure to achieve major quality improvements to become more competitive in the world marketplace, the term Concurrent Engineering (CE) is being used to express the desired environment to improve design quality. NASA’s formal definition of Concurrent Engineering is "the simultaneous consideration of product and process downstream requirements by multidisciplinary teams [NASA 1995]." CE can also be viewed as an implementation of the Total Quality Management (TQM) strategy. It can be described as a modern treatment of systems engineering which combines quality engineering methods in a computer integrated environment.

One of the extensions of concurrent engineering is the Integrated Product and Process Design (IPPD) [Schrage 1994, Mavris 1996, and Marx 1994].

IPPD provides design supporting techniques that help aircraft designers perform the trade-off studies to design better aircraft that meet the customer requirements by combining quality engineering and system engineering. It is a concurrent design with an integration of design and manufacturing and an optimization process that will consider design tradeoffs related to aircraft performance, utilization, and productivity. Design and manufacturing guidelines and constraints are established using the principle of Concurrent Engineering. We know that the freedom to alter designs decreases substantially when a design matures from a conceptual level to full scale production, see Figure 2.2.1. In addition, experience indicates that the earlier changes in design phases have greater opportunities to influence productivity. Hence, it is desired to incorporate productivity concepts early in an aircraft's design cycle.

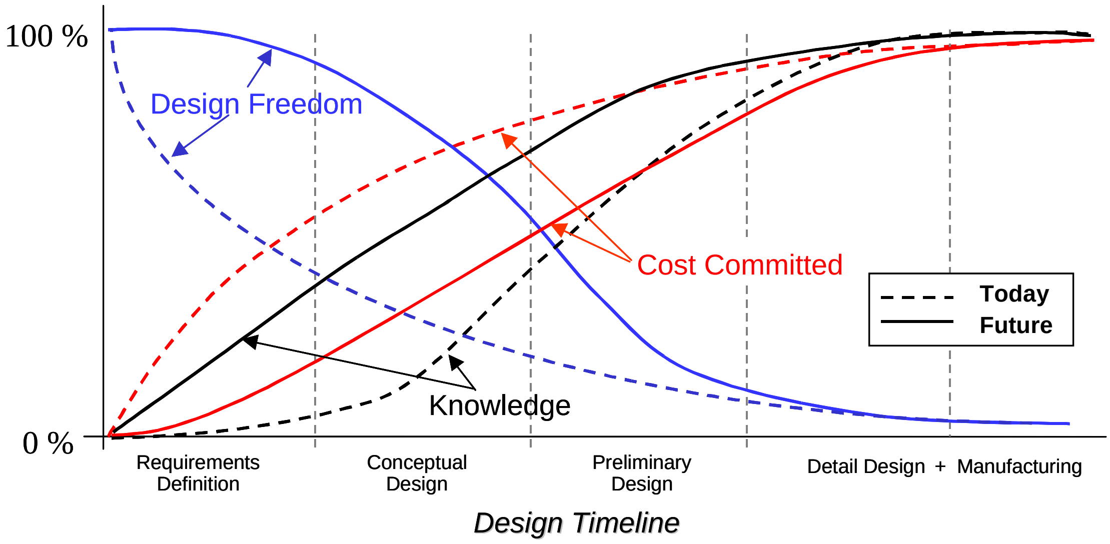



Figure 2.2.1 Relationship between Knowledge, Design Freedom, and Cost

A flow diagram for Integrated Product and Process Development during the various design phases is shown in Figure 2.2.2. The right half of the figure illustrates the decomposition activities from the conceptual system to components, to parts, to manufacturing process level. The small inner loops on the right half represent the product design trade iterations. The left half of the outer circle shows the process recomposition activities while the inner loops represent the process design trades. In past aircraft systems design, redesign was often required due to the incompatibilities between product design and manufacturing processes. Therefore, it is beneficial to have the ability to make parallel product and process design trades not only at the system level, but also at the component and at part levels. The IPPD environment shown in the center of process provides this capability.

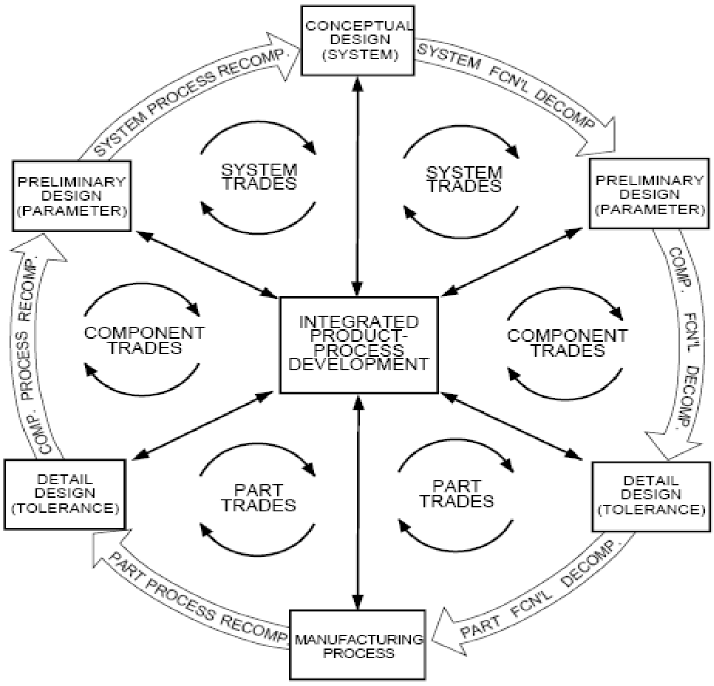



Figure 2.2.2 IPPD Process Flow

While Figure 2.2.2 provides the process flow, the methodology to implement IPPD is illustrated in Figure 2.2.3. It illustrates the interaction of the key elements necessary for parallel trades between product and process. There are four key elements: systems engineering methods, quality engineering methods, a top down design decision support process, and a computer integrated environment. The interactions for making parallel product and process design trades supports the top level elements. The methodology takes advantage of existing methods and tools in both products and processes. In this environment, system synthesis is achieved with Multidisciplinary Design Optimization (MDO) to generate feasible alternatives. These feasible alternatives are then evaluated with quality engineering methods for process robustness and decisions are made based on selection of the best alternative concept with certain criteria.

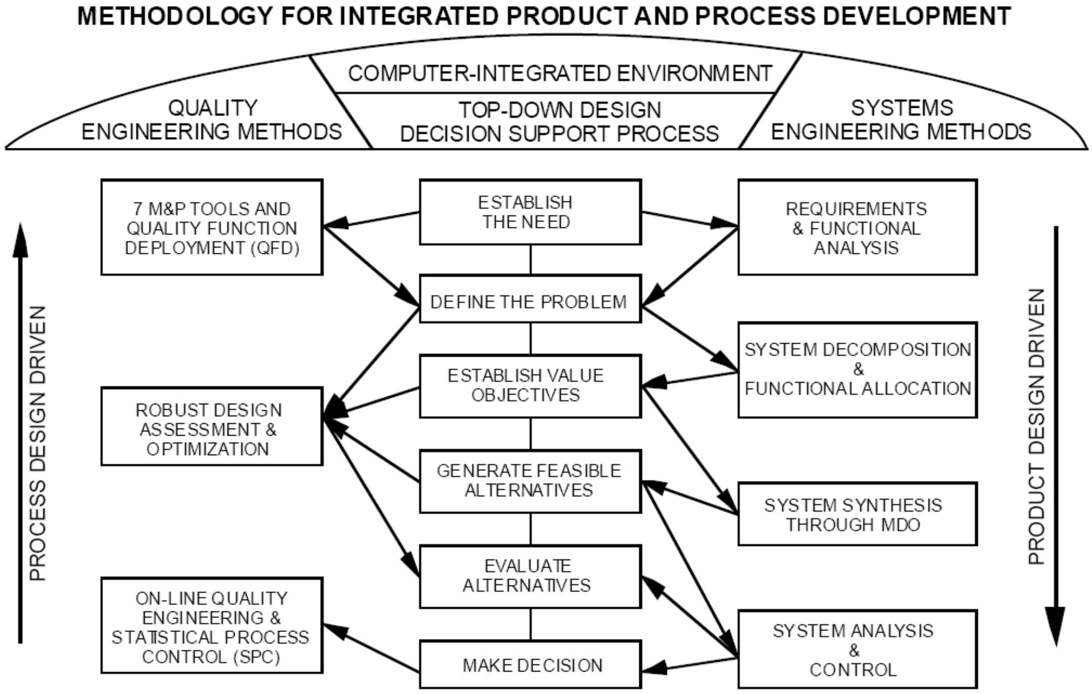



Figure 2.2.3 Methodologies for IPPD

As technologies evolve, design engineers begin to consider manufacturability concurrently in the design process [Dewhurst 1988, Dixon 1995, Shah 1995, Nederbragt 1998, and Feng 1999]. Some methods have been proposed for cost estimation [Dewhurst 1988, Winch 1989 Mileham 1993, Ou-Yang 1997], material process selection [Haudrum 1994, Mukherjee 1997, Giachetti 1998], and manufacturing engineering processes and technology [Kalpakjian 1995, Halevi 1995, Wong 1995, Feng 1999]. These research results create a foundation for integrated design and process planning at a conceptual level. However, the framework for integrated aircraft life cycle planning and decision support is still a new territory.

## 2.3 Knowledge Discovery

### 2.3.1 Knowledge-Based Engineering Systems (KBES)

Knowledge is the core of any engineering system. Several related research efforts addressed the need for Knowledge-Based Engineering (KBE).

Messimer [Messimer 1994] describe a materials and processing knowledge base and a model-based reasoning system developed to aid the selection and critique of composite materials and processes. The system is known as Composites Design and Manufacturing Assistant (CDMA), which was designed for composites producibility analysis.

The Lockheed Aeronautical Systems Company is also pursuing Knowledge-Based Design [Domeshek 1994]. The objectives of their efforts are the distribution of design knowledge in interactive advisory systems, the automation of multidisciplinary aspects of design, and the facilitation of IPD. The work includes use of a CAD system, a knowledge-based tool, a relational database, and a CAD/KB interface.

General Electric Aircraft Engines has also investigated the uses and applications of KBE [Williams 1991]. Their efforts are in different levels: the Knowledge-based Manufacturing Cost Estimator is focused on a limited class of compressors. The Assistant Cost Estimator will be applicable to the entire engine system. The Engine Development Cost Estimator is to predict engine development costs quickly and accurately to facilitate development cost trade studies.

McDonnell Douglas developed the Cost Benefits Analysis (CBA) model with the help of Arthur Anderson & Company [Jacobs 1996]. The CBA system is a PC-based model that estimates fabrication costs using an expert system. The expert system consists of a spreadsheet, a database, and a natural language interface. The database includes data for both metal and composite structures cost information. Studies were conducted at McDonnell Douglas using CBA to support HSCT design efforts to determine and analyze relevant cost issues.

Georgia Tech's Aerospace Systems Design Laboratory (ASDL) proposed a methodology for an aircraft producibility assessment [Marx 1995], utilizing a KBS for manufacturing process selection, that addresses both procedural and heuristic aspects of designing and manufacturing of a High Speed Civil Transport (HSCT) wing. A cost model is discussed that would allow system level trades utilizing information describing the material characteristics as well as the manufacturing process selections.

PACE has created knowledge-based engineering (KBE) applications which focus on supporting design with emphasis on preliminary design, products and systems configuration, and automatic document generation [Ward 2005]. KBE software from PACE adds a new approach to knowledge-intensive and multi-disciplinary design tasks by automating procedures characterized by frequent iterations and repetitive routines, thus yielding dramatic time savings and eliminating errors. Highly complex domain, process and algorithmic knowledge is identified, captured and formalized in a way that ensures this knowledge is not only automatically applied and readily accessible throughout the process, but also available for efficient reuse and distribution over department boundaries.

The development and growth of KBES presents an opportunity for the aerospace industry to replace the trend of increasing manpower with increasing computational power.

### 2.3.2 Knowledge Discovery in Data (KDD)

In today's environment, databases are widely used in almost every business and organization, which consequently results in an exponential growth in the amount of data stored. The overwhelming amount of data has easily exceeded the traditional human analysts' capacity of information digesting. The problem, known as "Data Overloading but knowledge starvation", is a common phenomenon across industries. Knowledge discovery in Database (KDD) is a key to resolve this dilemma, and may provide vital insight to the success of a business process. It seeks to intelligently analyze large amount of data in databases and extract previously unknown and useful knowledge from them [Fayyad 1996]. Since these data may come from all types of domains, such as business, financial, engineering, medical, etc. they contain valuable information that could be integrated within the organization strategy, and used to improve organization decisions. It is about leveraging artificial intelligence technology toward a strategic objective: competitive intelligence [Mena 1998]. Many companies and organizations have already gained beneficial insights into data that helped them to improve their business.

### 2.3.3 Data Mining

Data mining is a critical step in the process of knowledge discovery. It works on large amount of data and uses various data analysis tools to discover patterns and relationships in data that may be used to form higher level of knowledge or make valuable predictions. The field of data mining has made great strides during the 1990s, and continues to flourish into the new century [Han 2001].

Data mining is a multidisciplinary field, including research areas such as database technologies, artificial intelligence, machine learning, neural networks, statistics, pattern recognition, knowledge-based systems, knowledge acquisition, information retrieval, high-performance computing, data visualization, and so on.

The major thrust of data mining is the wide availability of huge amounts of data and the pressing needs on turning such data into useful knowledge. The knowledge obtained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration. The potential returns are enormous.

Data mining is a result of the natural evolution of information technology. The data handling has followed an evolutionary path with increasing functionalities, Figure 2.3.1. The early development of data collection mechanisms served as a foundation for later development of effective data storage and retrieval, and query and transaction processing. With numerous database systems offering query and transaction processing as a popular practice, data analysis has naturally become the next key role in the stage of data handling. Where data warehousing stores data in multiple heterogeneous data sources, and organizes them under a unified schema at a single site in order to facilitate management decision making.

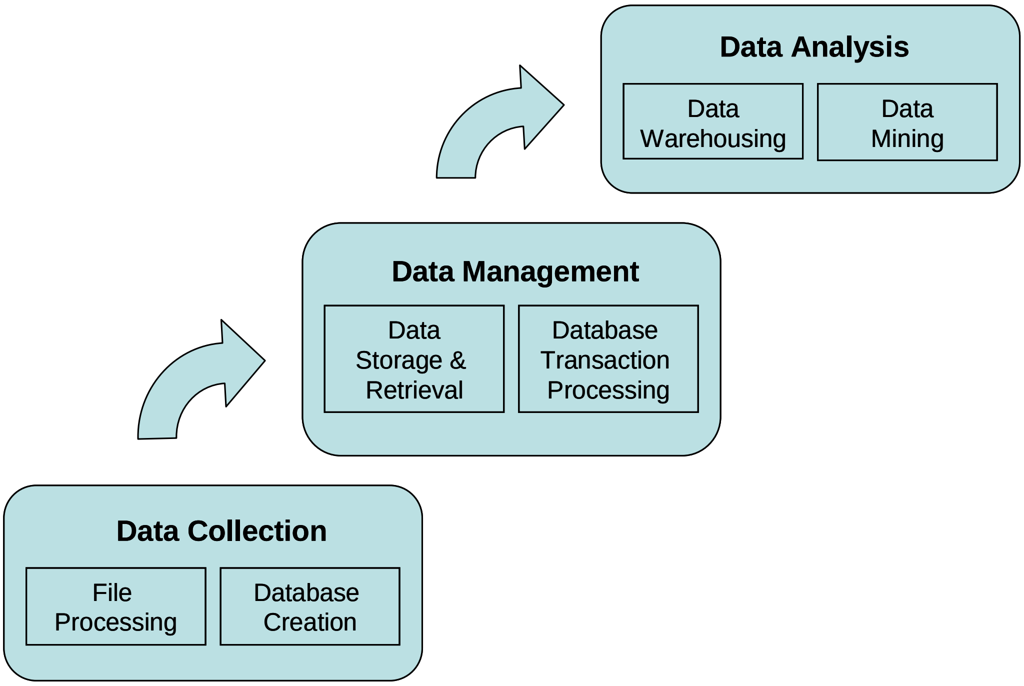



Figure 2.3.1 Evolution of Data Handling

Data mining is originally used in business sectors, such as retail stores, credit card companies, and banks, for customer relationship marketing (CRM) [Vagasi 2004, Xu 2002, Johson 2004]. Innovative organizations worldwide are already using data mining to locate and attract higher-value customers, to reconfigure their product offerings to increase sales, and to minimize losses due to error or fraud. Due to its strong potential, the techniques start to be extended into the engineering sectors [Pena 2000, Skormin 2002, Brence 2002], although they are still in an infancy phase.

Based on functionality, data mining can be classified into two categories: descriptive mining that discovers the general properties of the data in the database, and predictive mining that performs inference on the current data in order to make predictions.

Dealing with different purpose and based on information availability, data mining is commonly divided into two different types: supervised learning, where the mining results from the training samples are known in advance, and unsupervised learning, where the results from the training samples are unknown.

#### 2.3.3.1 Classification

Classification is a form of data analysis that can be used to extract models describing important data classes. Classification problems aim to identify the characteristics that indicate the group to which each case belongs. This pattern can be used both to understand the existing data and to predict how new instances will behave. For example, you may want to predict whether individuals can be classified as likely to respond to a direct mail solicitation, vulnerable to switching over to a competing long distance phone service, or a good candidate for a surgical procedure.

Data mining creates classification models by examining already classified data (cases) and inductively finding a predictive pattern. These existing cases may come from an historical database, such as people who have already undergone a particular medical treatment or moved to a new long distance service. They may come from an experiment in which a sample of the entire database is tested in the real world and the results used to create a classifier. For example, a sample of a mailing list would be sent an offer, and the results of the mailing used to develop a classification model to be applied to the entire database. Sometimes an expert classifies a sample of the database, and this classification is then used to create the model that will be applied to the entire database. Some common data classification techniques are decision tree induction, k-nearest neighbor classifiers, genetic algorithm, neural networks, Bayesian classification networks, case-based reasoning, fuzzy logic techniques, etc.

2.3.3.2 Decision Tree

Decision tree models are commonly used in data mining to examine the data and induce the tree and its rules that will be used to make predictions. Decision trees are grown through an iterative splitting of data into discrete groups, where the goal is to maximize the "distance" between groups at each split. A number of different algorithms may be used for building decision trees including CHAID (Chi-squared Automatic Interaction Detection), CART (Classification and Regression Trees), Quest, and C5.0.

One of the distinctions between decision tree methods is in the manner in which they measure this distance. While the details of such measurements are not discussed here due to space limitation, one can think of each split as separating the data into new groups that are as different from each other as possible. This is also sometimes called making the groups purer.

Decision trees that are used to predict categorical variables are called *classification trees* because they place instances in categories or classes. Decision trees used to predict continuous variables are called *regression trees.*

Decision trees handle non-numeric data very well. This ability to accept categorical data minimizes the amount of data transformations and the explosion of predictor variables inherent in neural nets.

Some classification trees were designed for categorical predictor variables. Continuous predictors can frequently be used even in these cases by converting each continuous variable to a set of ranges (binning). Some decision trees do not support continuous response variables for building regression trees, in which case the response variables in the training set must also be binned to output classes.

#### 2.3.3.3 Clustering

Clustering is the process of grouping the data into clusters so that objects within a cluster have high similarity between one other, but are very dissimilar to objects in other clusters. Unlike classification, one does not know what the clusters will be when he starts, or by which attributes the data will be clustered. Consequently, someone who is knowledgeable in the field must interpret the clusters. Often it is necessary to modify the clustering by excluding variables that have been employed to group instances, because upon examination the user identifies them as irrelevant or not meaningful. After one has found clusters that reasonably segment the database, these clusters may then be used to classify new data.

Clustering is different from classification. Clustering is a way to segment data into groups that are not previously defined, while classification is a way to segment data by assigning it to groups that are already defined.

The literature search conducted yielded a large number of references in the area of clustering algorithms. Based on this research review, clustering methods can be classified into the following categories.

**Partitioning methods**: They classify the n objects in the database into k groups (k<=n), and then they use an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general criterion of good partitioning is closeness, which means objects in the same clusters are “close” or related to each other, while objects of different clusters are “far apart” or very different. The definition of the distance is the key to the success of the reasonable partitioning, and sometime is difficult to determine. Two popular heuristic methods in this category are the K-means algorithm, where each cluster is represented by the mean value of the objects in the cluster, and the K-medoids algorithm, where each cluster is presented by the one of the objects located near the center of the cluster.

**Hierarchical methods**: A hierarchical decomposition of the given set of data objects is created by either a bottom-up (merge) approach or a top-down (split) approach, and iterated until termination condition holds. One major characteristic of hierarchical methods are the rigidity, i.e. a step (merge or split) can not be reversed. It means they cannot correct erroneous decisions. On the other hand, computation cost is greatly reduced by this rigidity.

**Density-based methods**: Most of the partitioning methods cluster objects based on the distance between objects. Such methods work well with round-shaped clusters and they have difficulties when discovering clusters with arbitrary shapes. The density-based methods achieve the clustering by the density (number of objects) of each cluster. The given cluster is to keep growing as long as the density in the neighborhood exceeds some threshold. Some common methods are Density-Based Spatial Clustering of Application with Noise (DBSCAN), which grows regions with sufficiently high density into clusters and discover clusters of arbitrary shape in spatial databases with noise, and Ordering Points To Identify the Clustering Structure (OPTICS), which computes an augmented clustering ordering in place of density-base clustering.

**Grid-based methods**: Where object spaces are quantized into a finite number of cells that form a grid structure. All of the clustering operations are performed on the grid structure. Some examples of grid-based methods are Statistical Information Grid (STING), where statistical information in each cell (such as the mean, maximum, and minimum value) are pre-computed and stored, and WaveCluster, which is a clustering approach using Wavelet transformation.

**Model-based methods**: It assumes a model for each of the clusters and attempts to optimize the fit between the given data and the mathematical model. Such methods are often base on the assumption that the data are generated by a mixture of underlying probability distribution. Statistical approach and neural network are two common approaches to this category.

## 2.4 Data Visualization

Data visualization is a vital aid in data presentation, and its importance to effective decision making cannot be overemphasized. Data visualization most often leads to new insights and success. Some of the common and very useful graphical displays of data are histograms or box plots that display distributions of values. Scatter plots can also be used to show two or three dimensions of different pairs of variables. The ability to add a third, overlay variable greatly increases the usefulness of some types of graphs.

Visualization works because it exploits the broader information bandwidth of graphics as opposed to text or numbers. It allows people to see the forest and zoom in on the trees. Patterns, relationships, exceptional values and missing values are often easier to perceive when shown graphically, rather than as lists of numbers and text.

The problem in using visualization stems from the fact that models have many dimensions or variables, but we are restricted to showing these dimensions on a two-dimensional computer screen or paper since humans cannot visualize high dimensional data as is. For example, we may wish to view the relationship between aircraft speed, weight, engine type, direct operating cost, sideline noise, etc. Data visualization techniques attempt to solve this problem by using clever representations to collapse *n* dimensions into two. Some of the techniques are discussed below.

### 2.4.1 Parallel Coordinates

A single row *i* in a data table with N attributes can be thought of as a point in an N-dimensional Cartesian coordinate system. For N>3 such configurations of points cannot be directly visualized; the method of Parallel Coordinates overcomes this limitation by arranging axes vertically, and spacing them uniformly across the plane [Unwin 2003, Edsall 2003, Inselberg 1985]. The parallel coordinates plot is the most straight-forward multivariate plot. The display is obtained by taking the dimensions as vertical axes thereby arranging them in parallel to each other, instead of using orthogonal axes as in the usual Cartesian graph. The individual data values are then marked off for each dimension onto the corresponding coordinate. The representation of a vector $x = (x_1, x_2, ..., x_n)$ is thus obtained by plotting $x_1$ on axis 1, $x_2$ on axis 2 and so on through $x_n$ on axis n. The resulting points on the axes are finally joined by connected line segments for each vector, yielding the parallel coordinate display of the data set.

A point in n-dimensional space is hence equivalent to a polygonal line through n parallel coordinates in this particular visualization method. From the structure of the resulting graph, one can draw conclusions for the relationship of the corresponding data values. A group of lines with a similar gradient can indicate that their data records correlate positively. Since each vector is represented in a planar diagram, each vector component has furthermore essentially the same representation. Another advantage of this visualization method is that the representation of all vectors in the same diagram means that a point pair wise comparison can easily be made. The drawback of this method is to effectively visualizing many variables with a large number of observations, and hence lines.

A sample plot is shown in Figure 2.4.1 with the data from a sample JMP file, Decathlon.jmp. In the example, the jump and field events are larger-is-better value, whereas the running events are smaller-is-better. By reversing the running events, the better values are consistently on the high side.

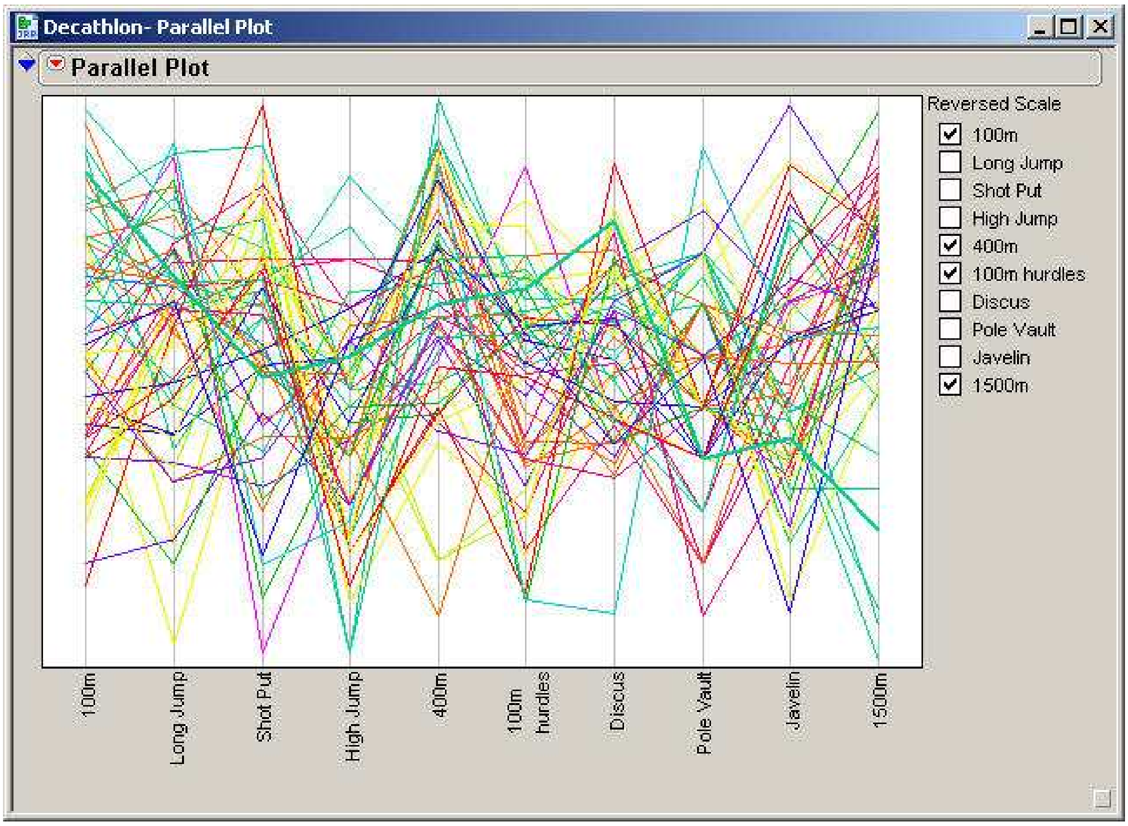



Figure 2.4.1 A Sample Plot of Parallel Coordinates

### 2.4.2 Star Plots

Star plots [Friendly 2003, Wu 1998, and Chambers 1983] are helpful to display multivariate records with an arbitrary number of variables. Each record is represented as a star-shaped plot with one ray for each variable. For a given record, the length of each ray is made proportional to the size of that variable. All variables in star plots are used to construct the plotted star figure. Instead, the star-shaped figures are usually arranged in a rectangular array on the page. It is somewhat easier to see patterns in the data if the records are arranged in some non-arbitrary order, and if the variables are assigned to the rays of the star in some meaningful order.

A star plot consists of a sequence of spokes at equal angles around a circle, and each spoke representing one of the variables. The data length of a spoke is proportional to the magnitude of the variable for the data point relative to the maximum magnitude of the variable across all data points. Line segments are drawn connecting the data values for each spoke. This gives the plot a star-like appearance and the origin of the name of this plot, Figure 2.4.2.

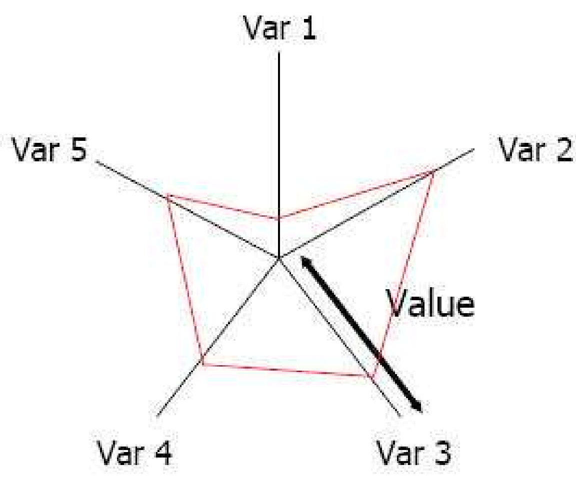



Figure 2.4.2 Demonstration of a Star Plot

Friendly illustrated a sample of star plots for the automobile data [Friendly 1991]. There are 12 variables to form the star, where the bottom spokes are related to size; the others relate to price and performance. In Figure 2.4.3, each star represents a car model. The dominant pattern is that the star symbols in the top rows have long rays on the top (good price and performance) and short rays on the bottom (small in size variables), which means smaller car has better price and performance than the heavier models.

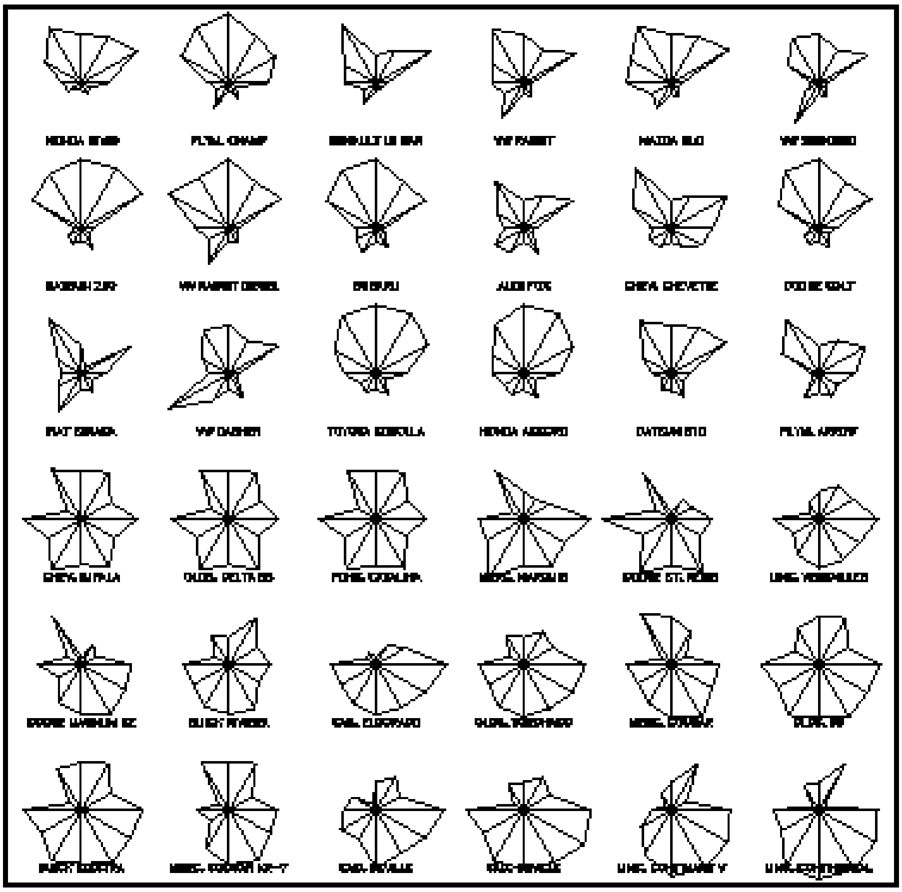


Figure 2.4.3 Star Plots of Automobile Data

In the star plot, one tends to see the configuration properties of the collection of variables represented for each observation, and that this perception is affected by the ordering of variables around the perimeter and by the arrangement of stars on the page.

Other arrangements might lead to noticing other features of the data, so it might be useful to try several alternatives.

### 2.4.3 Multidimensional Scaling

Like parallel coordinates, Multidimensional Scaling can also be used to visualize multivariate data [Huang 2006, Borg 1997, Cox 1994]. However, the original N axes and coordinates of points do not enter the visualization directly. Instead, a configuration of points is found in a space of lower dimension $M<N$, such that all inter-point distances match as closely as possible the original distances. A two- or three-dimensional embedding is an obvious choice for visualization; higher values of p can be useful for statistical analysis.

It might be helpful to imagine the process of multidimensional scaling in two dimensions as wrapping a surface - an elastic sheet - around points $\{u_i\}$ in the original high dimensional space, and taking $x_i$ as the projection of $u_i$ onto this surface. A non-linear mapping between the two configurations is established, and it is likely to be superior for purposes of visualization to rotating a rigid plane in the high dimensional space to find the closest fit to $\{u_i\}$, a procedure known as Principal Components Analysis [Pearson 1901].

### 2.4.4 Principal Components Analysis

The principal component analysis has been widely used in engineering fields [Zeng 2006, Anaparthi 2005, and Miskovic 2005]. In short, the principal component analysis takes the set of data points, and rotates it such that the maximum variability is visible. In another word, it identifies the most important gradients. The principal component analysis or Karhunen-Loeve transform is a mathematical way of determining that linear transformation of a sample of points in N-dimensional space that exhibits the properties of the sample most clearly along the coordinate axes. Along the new axes, the sample variances are extremes (maxima and minima), and uncorrelated. The name comes from the principal axes of an ellipsoid (e.g. the ellipsoid of inertia), which are just the coordinate axes in question.

By their definition, the principal axes will include those along which the point sample has little or no spread (minima of variance). Hence, an analysis in terms of principal components can show (linear) interdependence in data. A point sample of L dimensions for whose L coordinates M linear relations hold, will show only (L-M) axes along which the spread is non-zero. Using a cutoff on the spread along each axis, a sample may thus be reduced in its dimensionality [Bishop 1995].

The principal axes of a point sample are found by choosing the origin at the centre of gravity and forming the dispersion matrix

$$t_{ij} = (1/N)\sum[(x_i - \langle x_i \rangle)(x_j - \langle x_j \rangle)]$$

Where the sum is over the N points of the sample and the $x_i$ are the *i*th components of the point coordinates. The principal axes and the variance along each of them are given by the eigenvectors and associated eigenvalues of the dispersion matrix.

Principal component analysis has been used to reduce the dimensionality of problems, and to transform interdependent coordinates into significant and independent ones. An example used in several particle physics experiments is that of reducing redundant observations of a particle track in a detector to a low-dimensional subspace whose axes correspond to parameters describing the track. In practice, non-linearity of detectors, frequent changes in detector layout and calibration, and the problem of transforming the coordinates along the principal axes into physically meaningful parameters, set limits to the applicability of the method.

### 2.4.5 Scatterplot Matrix

An individual scatterplot does not generalize readily beyond two dimensions. For the visual representation of multivariate data, a more elaborate construct is needed: the matrix of scatterplots, where N given dimensions are projected onto N*(N-1) scatterplots. A scatterplot matrix is a collection of scatterplots organized analogously to a covariance matrix, with variable *i* plotted against variable *j* in the *i*th row and *j*th column of the matrix [Olive 2005, Clevelan 1984]. The diagonal plots can show the distribution of individual variables, or simply be placeholders for variable names. Individual scatterplots can reveal correlations between variables, for example, linearity, and the complete matrix can be useful for an initial exploration of a data set. However, the display becomes overwhelming if there are more than a few variables.

Although the scatterplot matrix is essentially limited to a collection of two-dimensional pair-wise comparisons, making it difficult to gain a real sense out of hyper-dimensional structures, this visualization method is still quite effective. In connection with other displays, it can act as a tool for zooming into two dimensions since for every two dimensions there is a scatterplot showing exactly their relationship.

An example of the multivariate scatterplot matrix is shown in Figure 2.4.4 with the same data set from Decathlon.jmp, and histograms showing distribution of each individual variable are plotted in the diagonal of the matrix.

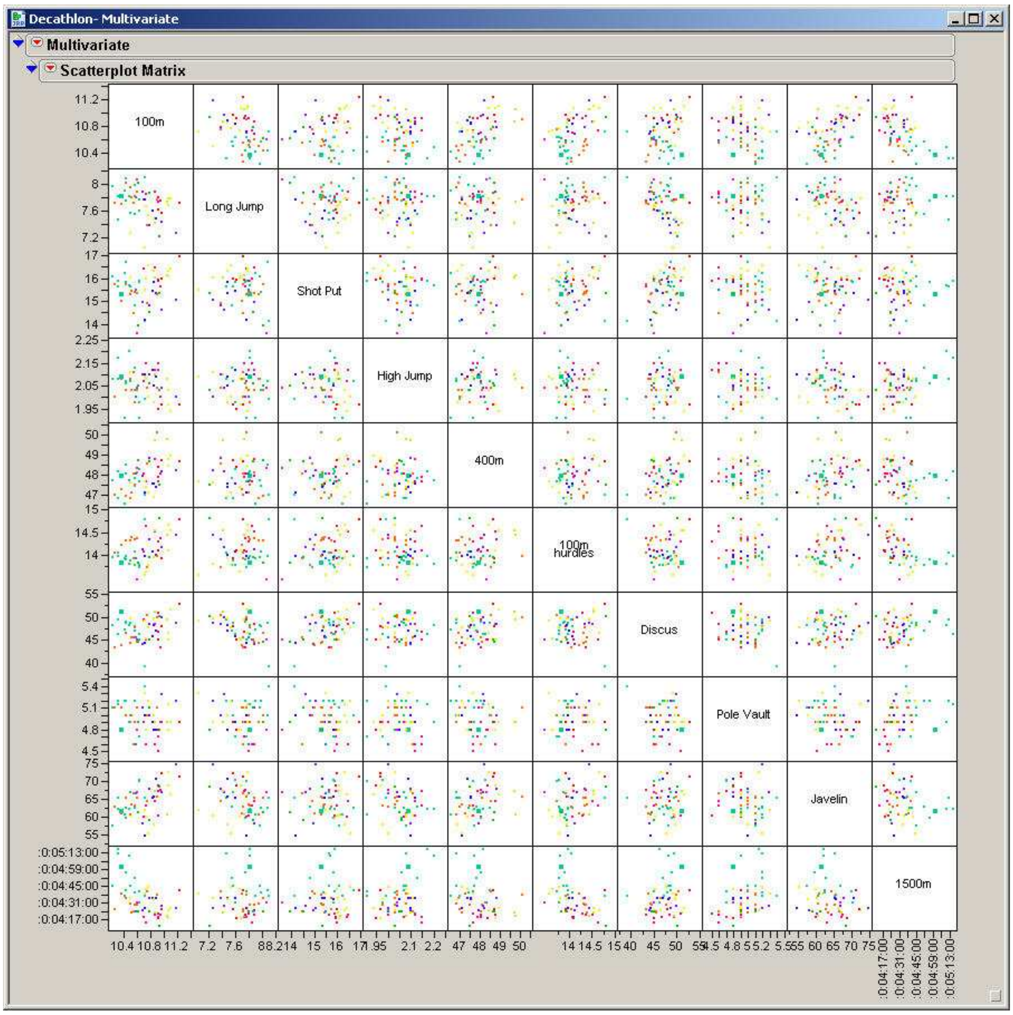



Figure 2.4.4 A Sample of a Multivariate Scatter Plot

The above figure records the performance of athletes in ten sport events, and visualizes the correlations on performance between each pair of sport events, which are listed in the diagonal cells and form the abscissas and ordinates of other plots. Each plot has an equal number of points, and the performance of an athlete in the events is represented by points across the plots. When a point in a plot is highlighted, the points associated to the same athlete are highlighted as well. The plots in the lower left section are the mirror image of those in the upper right section, since the plots are symmetric along the diagonal line.

### 2.4.6 Self Organizing Map

Self-organizing map (SOM) as a data visualization technique, which reduces the dimensions of data with self-organizing neural networks, has gradually gained its popularity [Shanmuganathan 2006, Barton 2006, Huysmans 2006, Jin 2004]. The way SOM is utilized to reduce dimensions is by producing a map of usually one or two dimensions, which plots the similarities of the data by grouping similar data items together. Therefore, SOM accomplishes two things in data visualization, reducing dimensions and displaying similarities, which are very useful in data clustering. Therefore, SOM is also a data mining technique.

The SOM uses an algorithm to visualize and interpret large high-dimensional data sets. Typical applications are visualization of process states or financial results by representing the central dependencies within the data on the map.

The map consists of a regular grid of processing units, "neurons". A model of some multidimensional observation, eventually a vector consisting of features, is associated with each unit. The map attempts to represent all the available observations with optimal accuracy using a restricted set of models. At the same time, the models become ordered on the grid so that similar models are close to each other and dissimilar models far from each other.

Fitting of the model vectors is usually carried out by a sequential regression process, where $t$ = 1, 2... is the step index: For each sample $\mathbf{x}(t)$, first the winner index $c$ (best match) is identified by the condition

$$\forall i, ||\mathbf{x}(t) - \mathbf{m}_c(t)|| \leq ||\mathbf{x}(t) - \mathbf{m}_i(t)||.$$

After that, all model vectors or a subset of them that belong to nodes centered around node $c = c(\mathbf{x})$ are updated as

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{c(\mathbf{x}),i}(\mathbf{x}(t) - \mathbf{m}_i(t)).$$

Here $h_{c(\mathbf{x}),i}$ is the neighborhood function, a decreasing function of the distance between the *i*th and *c*th nodes on the map grid. This regression is usually reiterated over the available samples.

## 2.5 Information Management

Information management is becoming more and more important, as more and more information generated with the help of information technology and World Wide Web. To efficiently store, and retrieve information is the primary task of information management. Zhao proposes an information modeling approach for the conceptual process planning [Zhao 2004]. Wang demonstrates a holistic approach to the knowledge management view for companies in the manufacturing industry planning to create competitive advantage for itself [Wang 2004].

MIT developed a digital management environment, named DSpace. As a digital archive system, it captures, stores, indexes, preserves and redistributes an organization's research material in digital formats. It is similar to a traditional database, but has the capability to support many types of digital formats. On this sense, it is more like a file system with indexing capability. Research institutions can use DSpace for a variety of digital archiving needs -- from institutional repositories to learning object repositories or electronic records management, and more.

# CHAPTER III

# RESEARCH QUESTIONS & HYPOTHESES

To create better designs, it is critical to develop a methodology to discover, retain, organize, and present knowledge throughout all phases of the aircraft life cycle. This methodology considers the aircraft design, production, operations, and ground support. The definition of this need leads to a series of research questions that this thesis attempts to resolve.

## 3.1 Research Questions

1. How does one obtain knowledge from existing data in aircraft life cycle activities? This can be divided into two sub-questions:

1.1 How does one generate representative elements from data to clarify the process?

1.2 What mechanism is needed to investigate large amounts of data?

Once knowledge is obtained and in order to make the knowledge useful, an environment is required to store this knowledge and to present it to the decision maker when it is needed. Therefore, a set of secondary research questions are posted:

2. How does one manage explicit knowledge efficiently throughout the aircraft life cycle in order to make it easily accessible?

3. How does one present knowledge effectively and transfer it to design decision maker?

## 3.2 Hypotheses

The hypotheses formulated to answer the research questions are as follows:

I. Knowledge can be obtained from existing data in the aircraft life cycle with the help of features and hierarchical investigation (Research Question #1)

  - Feature exploration algorithms can extract and generate features, which transform the data, and guide the knowledge discovery process (Research Question #1.1)
  - Hierarchical investigation can be used to systematically examine the data (Research Question #1.2)
    - Hierarchical composition simplifies the data by displaying a holistic view with details removed
    - Hierarchical decomposition performs a detailed multi-step investigation with focus on particular issues

II. Through the proper definition and classification of a hierarchical system structure, knowledge in the aircraft life cycle can be efficiently managed in a customized digital repository system (Research Question #2)

III. Knowledge-enabled visualization models offer a better knowledge presentation to the decision maker at the proper level (Research Question #3)

## 3.3 Key Technical Challenges and Solution Alternatives

- Effective knowledge discovery in large aircraft life cycle related databases can be accomplished using the following options:
  - Option 1: Traditional database query with domain knowledge
  - Option 2: Standard data mining techniques
  - Option 3: Feature-based hierarchical knowledge discovery

Option 1 provides an ad-hoc approach to investigate information in a database with existing knowledge in the domain. Although it is very flexible, it requires extensive involvement of domain experts and lacks sufficient theoretical guidance for the direction of the exploration. Option 2 offers a typical approach to explore large amount of data and find patterns. However, standard data mining techniques were originally developed for business purposes, and are not well suited for applications in aerospace engineering, where the structure and the characteristics of data are specialized to serve a particular purpose. In those data, a large amount of domain information is a prerequisite for users and hence missing from the data. This presents a big barrier for the automated data mining programs. Therefore, the data mining approach is incapable of efficiently generating meaningful results in aerospace engineering data. Option 3 provides an architecture that is designed to resolve the dilemma by the means of feature-based hierarchical knowledge discovery. It first explores the features with a combination of mathematics and the expert domain knowledge through a structured feature generation algorithm, and then uses the created features to investigate the data with a hierarchical system recomposition and decomposition approach. As we can see, option 3 offers a structured approach that is lacking in option 1, and overcomes the obstacle in option 2 with features and system hierarchy.

- ❖ Management and access of knowledge throughout the aircraft life cycle can be achieved using the following options:
    - Option 1: Product Life cycle Management System (PLM)
    - Option 2: Digital repository systems

Option 1, Product lifecycle management (PLM), is a process of managing a product from its cradle to its grave. It enables an enterprise to innovate and manage its products and related services throughout the entire business lifecycle effectively and efficiently (Wikipedia 2006). The product lifecycle goes through many phases and involves many professional disciplines and requires many skills, tools and processes. PLM is more about managing descriptions and properties of a product through its development and useful life, mainly from a business/engineering point of view. There are a few powerful commercial applications, such as ENOVIA from IBM, and TeamCenter from UGS. Option 2 stores resources in machine-readable format and accessible by means of computer. It is a concept derived from digital library, which digitizes catalogues, periodicals and books, and provides indexes, abstracts and references. Instead of being bundled with the product design and analysis environment as in PLM, digital repository systems are mostly standalone systems that are flexible and easy to customize and can be extended to suit special needs. Option 2 utilizes open source and freely available architectures and creates applications that are freely distributable, which fits the affordability and portability concern in academic research. There are some open source digital repository systems, such as DSpace developed by MIT [DSpace 2005] and Fedora by Cornell [Fedora 2005]. Combined with a hierarchical system structure, the knowledge developed throughout the aircraft life cycle can be stored and managed efficiently in a customized digital repository system.

- Knowledge presentation to empower design decision making can be accomplished using the following options:
  - Option 1: Ad hoc approaches: tables, charts and trends
  - Option 2: Visualization tools: OpenMind [Matchware 2007], Knowledge-based Multimedia Morphological Matrix, etc

Option 1 utilizes traditional presentation methods to communicate with listeners, where knowledge is displayed in a static manner, and the structure of the knowledge can be hard to present even with a carefully designed scheme. Option 2 creates a clear hierarchy of the aircraft life cycle knowledge in order to facilitate design decision making. With the help of visualization tools, such as OpenMind and a knowledge-based multimedia morphological matrix, knowledge can be organized in a well structured way, and presented completely and dynamically through modern techniques, such as video, audio, webpage, animation and simulation. Thus, presentations are more informative and do not rely exclusively on static information, but utilize more channels, making knowledge transfer a smooth, effective, and even delightful experience.

# CHAPTER IV
# TECHNICAL APPROACH

## 4.1 Knowledge Engineering for Aircraft Life cycle Design Decision Support

A framework of knowledge engineering has been established to systematically process knowledge to facilitate aircraft life cycle design decisions. Aircraft life cycle creates a complex system of systems. Aircraft design, airframe and components manufacturing, aircraft operations, and maintenance form the interactive links of the entire aircraft life cycle. Each of the phases also consists of multi-level complex systems that need to be designed and integrated. For example, the aircraft design is composed of aerodynamic design, structural design, propulsion design, and so on. In addition, the process to create a model and prototype for production evolves through various stages of design, from conceptual design, to preliminary design, to detailed design. There are multiple goals and constraints throughout the life cycle. Some goals include economics, performance, manufacturability, availability, reliability, and maintainability. The constrains include but not limited to availability and readiness of advanced technologies, and government regulations on noise and emission. Enormous amount of knowledge is needed to achieve a successful design that satisfies all design goals.

Where do we get the knowledge? Is it from experience, from books and theories, or from trial and error? The root source of knowledge is the information and data that we obtained, and the process of understanding them. Observations from past activities, such as experiments, are processed with a human brain or computer to gain experience that then can be further developed and generalized to create theories. Different forms of knowledge are stored in different places for later use. Just as experiences are stored in a human's mind which later might be hard to recall or be transferred to other persons, theories are usually found in text books and reports and can be hard to find when the information is needed. The purpose of this thesis research is to find an efficient way to understand past information, create knowledge, manage knowledge and information to make them available when needed, and create a method to present it in a straightforward format to decision makers for easy understanding.

With the help of information technology, the aerospace industry has accumulated a huge volume of data, and the size of information is growing continuously. Finding valuable knowledge from the overwhelmingly large data size is a challenge to investigators. Based on all the activities throughout the aircraft life cycle and the consideration of performance, economics, and government regulations, the framework described in this thesis extracts knowledge from existing data, manages it in a structured manner, and provides valuable insights to the designer when needed. The framework is composed of four top level components as displayed in Figure 4.1: knowledge discovery, knowledge integration with data configuration control, knowledge management, and knowledge presentation. All these components are within the knowledge engineering framework. In this framework, knowledge and information flow between the components.

Design for aircraft life cycle can not be accomplished in a single activity, but rather in a variety of activities surrounding the aircraft. These activities include design engineers creating the aircraft concept based on customer requirement, manufacturers formalizing the production processes and building the aircraft. Then the airlines, military, and other entities acquire and operate fleets of aircraft to achieve certain goals. Maintenance engineers provide support to aircraft and repair them throughout the operational life of the aircraft to make sure that aircraft are in good shape and are safe to fly. The goal of the proposed research is to establish a set of processes to generate knowledge from existing data and propose a method to provide knowledge to decision makers, in a way that decisions can be made with the aid of better knowledge about the problem.

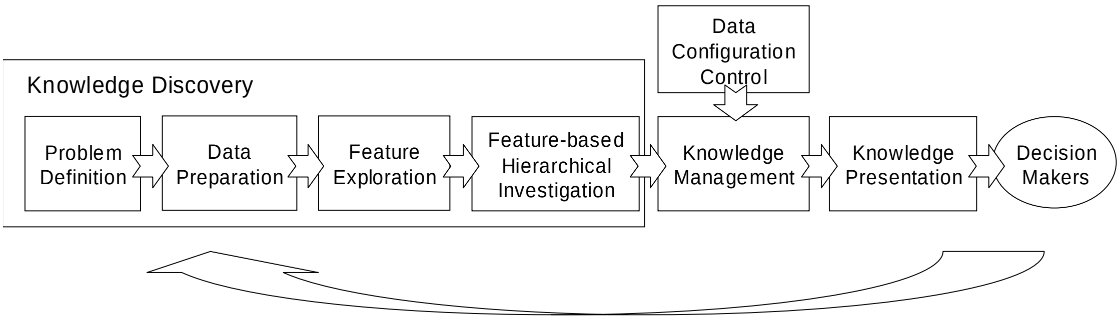



Figure 4.1.1 Knowledge Engineering for Aircraft Life cycle Design Decision Support

In this framework, knowledge is created by the knowledge discovery component with proposed knowledge mining methodology, which applies a feature generation approach to effectively capture and generate features to represent information at higher levels, and improves the process of knowledge mining. Data are reorganized and transformed into a concise but information rich format with the guidance obtained from features. Then, a feature-based hierarchical system investigation is carried out to seek knowledge from the reconstructed dataset.

After the knowledge is created with related information, it is then stored with the knowledge management component, which maintains the knowledge in an organized structure and has the ability for geographically distributed users to search the content via internet and access the knowledge remotely.

To increase the reusability of the knowledge, it is important to retain the underline information related to the knowledge at a proper level while knowledge is created so that knowledge can be easily updated when new data are attached to the existing data. It also eases the knowledge creation process since it provides the capability to reuse a portion of existing data if the conditions are only slightly changed. A standardized data configuration control process is proposed and implemented to maintain the integrity of the knowledge. One would first create a set of standard data and information templates based on the investigation of the process of knowledge generation. Then one would use knowledge templates as a data structure for incorporating information as the model creation process progress. A sample implementation of this approach is demonstrated.

Finally, to support various levels of decision making, the knowledge can be presented to the designers in easy-to-understand forms in the knowledge presentation component where knowledge is enhanced with organized and information rich visualization models. The proposed approaches can clarify the structure and content of knowledge and offer better understanding and interpretation of the knowledge at the proper level.

Furthermore, the feedback from the designers can be used to help the knowledge discovery process and find more knowledge that is relevant.

This framework is flexible by enabling modular processes, which means that the applications in each of the components are modules. The tools used in the thesis are for demonstration purpose, and other similar tools can easily be plugged in to replace the current ones without much modification. It brings the flexibility to fit users' special situation.

A taxonomy of the information process illustrates the proposed process (Figure 4.2). Information can be treated in various ways and serve different purposes based on the taxonomy. Some sample information categories are illustrated.

When we have databases with a large volume of data, regardless of its database management environment (Microsoft Access, Oracle, DB2, Excel, raw text file, and so on), feature-based hierarchical knowledge discovery can be applied on the data. The process goes through feature generation, hierarchy definition, data mining, to extract valuable knowledge, and present knowledge to designers for decision making. In parallel, discovered knowledge can be stored and managed in the knowledge management environment.

When we have surrogate models, such as response surface equations, neural nets, and genetic algorithms, the data configuration control process can be utilized to maintain the integrity of the model. The process employs standard templates predefined based on the general flow of model creation, and related information is kept for future reference. In addition, the complete information can be kept in the knowledge management component for remote access.

In the information era, many multimedia data are provided to designers in various formats, such as video, audio, graphic, webpage, and so on. In the process of alternative selection, with knowledge-based multimedia morphological matrix, the data can be linked to a specific technology, and organized in a preferred manner, according to the alternatives listed, to give designers more details on the technology, and provide a direct support on decision making.

In aircraft life cycle, there is lots of disciplinary specific information in different phases of the cycle, such as engine flow path, aircraft geometry, and so on. A visualization platform is implemented with hierarchy for easy information review, and gives designers a direct and organized view of the related information.

In conjunction with all the processes of knowledge discovery and presentation, the knowledge management environment provides an underline support of the knowledge storage with knowledge structure definition, where knowledge is categorized into proper hierarchy of the aircraft life cycle.

The above section provides a high-level overview of the proposed research work, and each of the components is demonstrated in detail in the following sections.

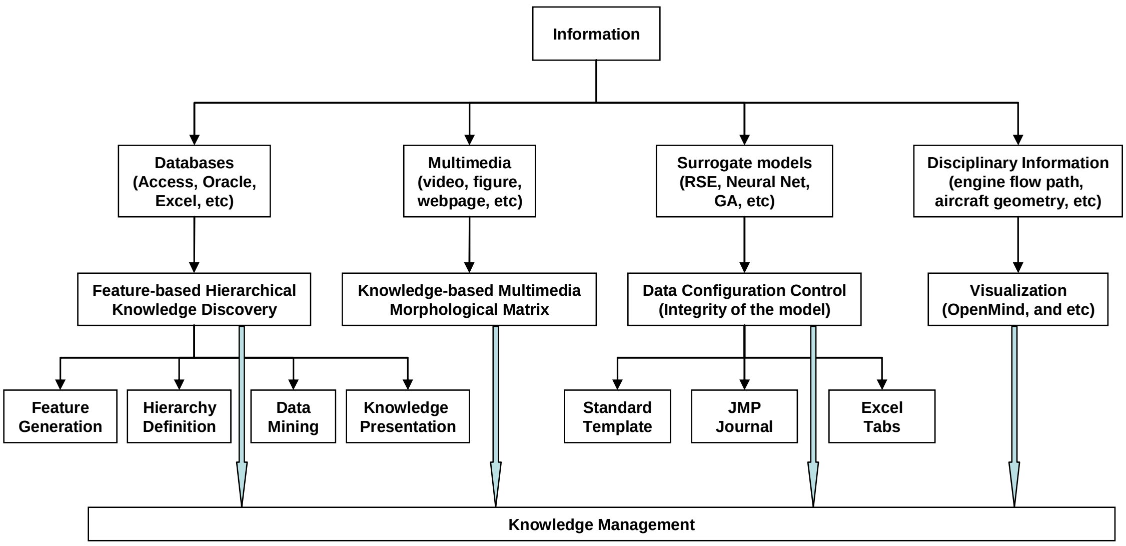



Figure 4.1.2 Taxonomy of the information process flow

## 4.2 Knowledge Discovery for Aircraft Life cycle Design Decision Support

As mentioned early in the motivation section, in the aircraft life cycle, more and more data have been gathered with improving capability from information technology infrastructure. In addition, a systematic approach is needed to effectively discover the knowledge hidden in the mountains of data that exist and are growing in the aerospace industry, and make it valuable to support aircraft life cycle decision making.

Current available algorithms of Knowledge Discovery in Database (KDD) are mostly generic approaches developed for business sectors. Although a trend to apply KDD in the engineering field starts to appear, it is still in the infancy phase. Based on the author's knowledge, there are very few attempts to apply KDD in the field of aerospace industry because of the complexity of the system. The popular KDD algorithms, such as Classification, Decision tree, K-means clustering, and Self-organized Maps, often fails on extracting useful knowledge from such systems. The problem is that the given engineering data themselves are not suitable for traditional data mining. A novel approach in knowledge discovery is proposed here to tackle database in complex systems of aerospace industry.

There are two common characteristics in aircraft related databases: Huge number of records, and only a small number of fields. The mountain of records is a presentation of large amount of tasks and activities on aircraft systems. The relatively small number of fields is the result of limited information sources. It is a difficult task to dig out valuable knowledge from restricted information channels. On the other hand, the huge amount of records also post capacity and efficiency barriers on knowledge mining. The proposed methodology is specially designed to deal with the two distinct characteristics of the aircraft related databases. A bottom-up approach is used to generate new fields based on higher level abstractions, and the amount of records will be significantly reduced due to the abstraction. Once some high level knowledge is discovered, a top-down approach is used to distill detailed knowledge through system hierarchy.

The steps of the proposed methodology are described in Figure 4.2.1. Starting from the bottom of the graph, one will first define the problem, which is to gather requirements, and clarify the goal for the knowledge discovery. Then the data are prepared for the KDD from the given database, which include compacting, cleaning and redundancy removing. In the next step, one will explore possible features, which are the key elements, and they may or may not exist in the given dataset, but are critical on presenting data for the mining process. It includes feature generation, identification and validation. In this step, we propose a methodology to generate features hierarchically. After we have a hand full of features, we need to reorganize the data into a new dataset based on the features generated, where the data structure could be very different compare to the original dataset. Then one will proceed to perform feature-based knowledge mining using certain data mining programs. The results will be visualized and interpreted. Irrelevant information can be filtered out at this point. Finally, valuable knowledge is organized, customized and provided to facilitate decision making.

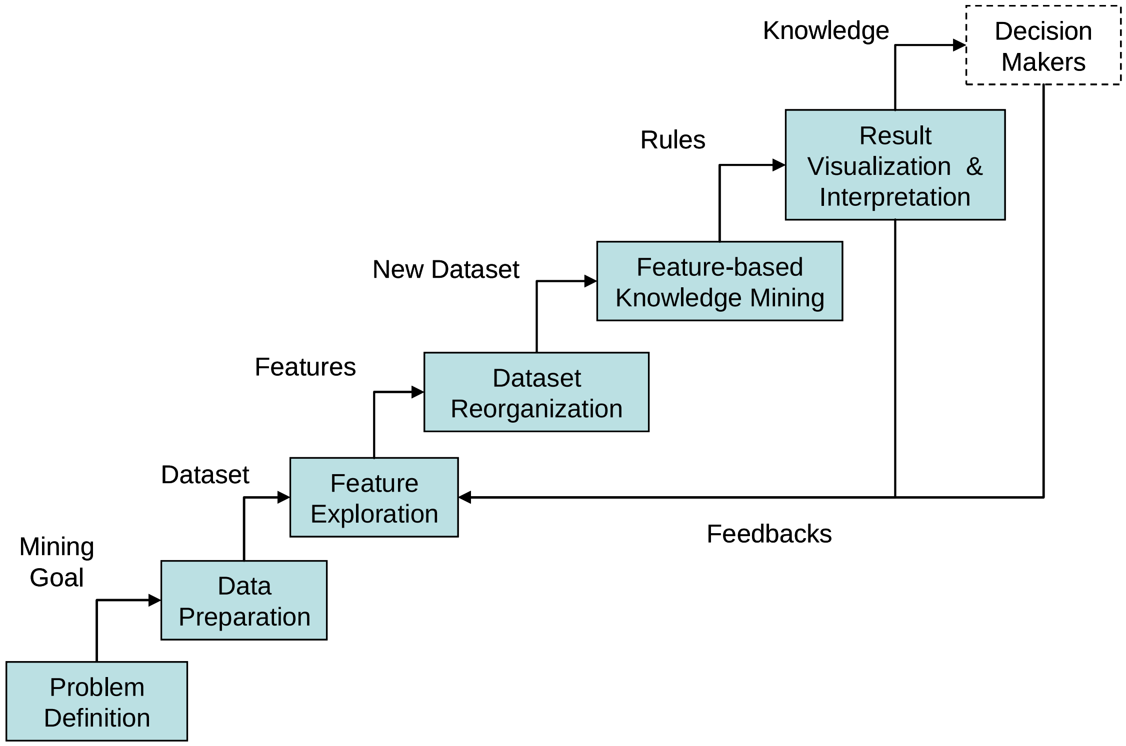



Figure 4.2.1 Feature-Based Hierarchical Aircraft Knowledge Discovery

### 4.2.1 Problem Definition

To effectively perform knowledge discovery in database, one needs to clearly define the overall goal, which is the foundation of the whole process, and provides a guideline in every of the later steps to make sure that there is no deviation in the overall direction. For example, in aircraft design, what are the key factors to design an aircraft, which is affordable to develop and produce for the manufacturer, and meet airline expectations in terms of loading capability, speed, and range? In airline operations, what are the key elements to operate the fleet effectively, which increase the revenue and reduce the operating cost? In aircraft maintenance, what are the suggestions and recommendations on improving airline unscheduled maintenance operation?

Evidently, the problem can quickly become very complex due to the nature of the system. Moreover, the customer goals are often ambiguous and need to be converted into detailed direct metrics. For example, an airline's desire to lower their operating costs could be translated into shorter flight routes, better flight schedules to accommodate maintenance, or a better arrangement of flight crew members. Quality Function Deployment (QFD) offers a structured means to translate the 'Voice of Customers' into the 'Voice of Engineers' through systematic brainstorming [Kusiak 1993].

On the other hand, to clearly define the goal might be a hard task in the beginning, since one may not be familiar with the given database and may be not clear what to expect and which direction to go. In that case, we could use some help from QFD to clarify the definition.

Through the QFD process, we can identify the appropriate target of the mining process, and then we can prepare the data based on the mining goal.

### 4.2.2 Data Preparation

There are various formats of data, such as structured data in databases, paper prints, microfilms, row data in files, program input and output files, charts and figures, pictures, audio and video media files. For the scope of this research, focus is on the numerical and categorical data stored in databases. Techniques on extracting information from multimedia data, such as voice recognition and image processing, are not discussed in this thesis.

Since the data directly obtained from the database are sometimes not directly usable. The reasons are as follows:

- Heterogeneity: Data may come from different databases, with different formats and data structures.
- Irrelevance: Not all data stored in the databases are relevant to the mining goal defined.
- Errors: Not all data stored in the databases are right, some of them may be entered incorrectly because of human typos or converting errors.
- Missing fields: Not all fields are populated in the database, some of the fields are left blank due to lack of information
- Redundancy: Certain records are entered more than once due to various reasons.
- Amount: The data reside in the databases may have a huge volume, which might affect the analysis efficiency.

We will need to pre-process the data to make them easy to use and with least amount of errors for further analysis. The techniques, which we are applying for data preparation, are as follows:

*Converting*: Based on the data mining goal, we can define a special data structure for the analysis, and the data from various sources can be converted into a common format. We also need to decide the data platform that will be used to store the resulting dataset. This is based on the considerations on the available data source, the level of effort on the converting, the size of the data, and the handling capability and specification of the analysis program using. For example, we could use tab delimited ACII file, Microsoft Excel, Microsoft Access, Oracle, DB2, JMP, etc. Wrapper programs can be written to process the data for different databases, and conversion can be done automatically once it is setup.

*Filtering*: The data mining goal has defined what we want out of the dataset. Therefore, data not related with the problem can be eliminated to improve efficiency. It will also reduce the amount of the data we need to process. Queries with specified criteria can be executed to filter out unnecessary records.

*Cleaning*: To clean the dataset requires domain specific knowledge, since one will need to figure out what records are not making sense. It could be a difficult task, since the data volume is large. However, the benefit from it could also be large with potential outliers, or even worse, misleading data removed in advance. To handle large volume, certain expert systems can be used to screen the data, and remove erroneous records.

*Leveraging*: It is common in real world that the data sources are not complete, relevant fields are missing information in certain records. To deal with this situation, there are two options. One is to neglect the related data records, which is an easy way to handle it, but it may cause inaccuracy if critical data are removed. The other option is to keep the records, and try to fill in the blank fields by creating information using other similar records as references so that the general trends in the whole dataset will not be disturbed. One way to fill in the blank is to calculate the average from similar records, and use it as the value. If no similar records exist, some regression methods, such as interpolation or extrapolation, may be applied.

*Compacting*: The process of removing redundant records has two purposes: one is to compact the dataset, the other is to reduce the influence of the records in further analysis. This seems could easily be done, but we need to proceed with cautious.

We do not want to remove different records that are very similar. For example, two maintenance records at the same time and location serving the same aircraft, and with the same service code, but they have different care remarks: one is “REPLACED NUMBER 2 MAIN WHEEL ASSEMBLY”, and the other is “REPLACED NUMBER 3 MAIN WHEEL ASSEMBLY”. Therefore, they are representing two difference service, and should not be considered redundant. The only time they can be treated as same service is taking wheel assembly as one type of the tasks.

On the other hand, we do not want to keep identical records that look different. Taking the previous sample, if one has remark as “REPLACED NUMBER 2 MAIN WHEEL ASSEMBLY”, and the other one has “RPLD NBR 2 MAIN WHEEL ASSY”, then they are exactly same, and might just be entered twice by different technicians.

*Trimming*: It is limiting the range of data, so that the further analysis will have more focus on certain issues. It also reduces the running overhead of queries, since it may significantly drop the amount of data volume. For example, if we want to investigate the aircraft maintenance service behavior in the winter, we can take data from October to next March, and cut all the other data out off the dataset. If we want to explore the unscheduled maintenance, then we can delete all the scheduled maintenance records from the dataset.

Now we have a data set from the original database according our goal, then we can proceed to explore features in the dataset.

### 4.2.3 Feature Exploration

To most efficiently gain knowledge from a data set, we need to identify a set of classification and evaluation criteria, called features. Features contain essential characteristics of the data, and can bring knowledge much closer to us from the mountains of data. With features explored, we can optimize the amount of effort and maximize the amount of useful output. Due to the importance of this step, the feature exploration process is discussed in a separate section (chapter 4.3).

### 4.2.4 Dataset Reorganization

After the features are generated and explored, a new dataset with different content need to be recreated based on the features. New tables are built based on grouping features, and new columns are added based on evaluation features. This is a reorganizing process to the given data. Distilled with the features, the new dataset will be more compact in size and has more information in content.

Once the new dataset is generated with the features, we have a better grasp of the data we are facing, and can investigate the dataset with those features. Since the feature generation and identification itself is a mining process, the new dataset is at a higher level of information abstraction, and much closer to the knowledge we are seeking.

### 4.2.5 Feature-Based Knowledge Mining With Hierarchical Analysis

Aircraft system is a complex system of systems, and consists of multiple levels of subsystems, which themselves are complex systems containing subsystems. Feature exploration clarifies the content, and we now need systematic procedures to simplify the content so that it can be focused at a proper level, which contains valuable information in a preferable size.

Two hierarchical analysis approaches are proposed to perform knowledge mining in complex aircraft system databases, where traditional data mining techniques are incapable to generate valuable knowledge. One is a bottom-up approach, which is to sum things up in groups from bottom and look at them together (this is simplification by integration and abstraction); the other is a top-down approach, which is to break things down into subcomponents from top, and pick some smaller portions with special focus (this is simplification by division and concentration). In another word, we first integrate lower level systems into a higher-level system with a system composition process, which reduces the complicity of the system, and gain overall understanding of a system. Once some abnormal phenomena are spotted at a higher system level, we can use a system decomposition process, which is a top-down approach to break down a complex system into smaller subsystems, and obtain greater focus by reducing the scope of analysis to investigate each of the sub level systems, and find potential causes of the abnormity. One example is shown in Figure 4.2.2 for aircraft support and maintenance system analysis.

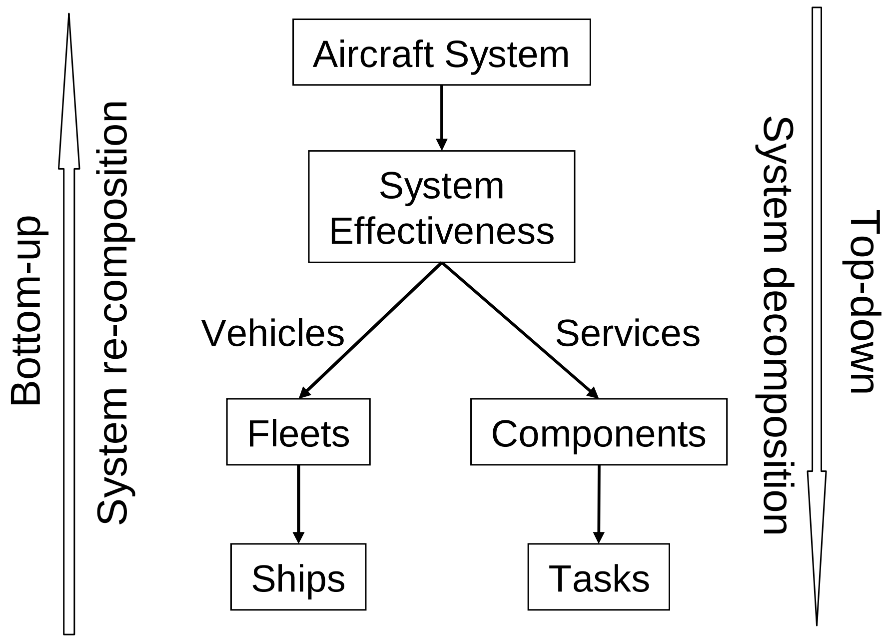



Figure 4.2.2 Hierarchical Analysis for Aircraft Support and Maintenance System

There are millions of service records in an aircraft support and maintenance system, and each record is a log containing the time, duration and location of an individual service job performed on a ship which is a specific aircraft. We use the word task to represent the same type of individual service jobs. Ships and tasks are the fundamental elements of the aircraft maintenance system, and there are thousands of tasks to support hundreds of ships. They represent two interrelated aspects of the system, vehicle aspect and service aspect.

Bottom-up approach is to synthesize system from the bottom levels according to these two aspects. We group all the activities by tasks and utilize the explored features, such as average task man hours, service frequency, and total task man hours, then we have performance measures for each of the tasks; certain tasks, based on the aircraft component they served, can be grouped into a higher level task category. Combined with vehicle information, we can compare service effectiveness at component level.

In this approach, the aircraft components are defined by the ATA code. The ATA code, also called ATA specification 100 code, is set forth by the Air Transport Association of America (ATA) to code the structural and functional components common to most aircraft. It is a four-digit code, of which the first two digits reference a major structural/functional system or component grouping on an aircraft, and the third digit references a subsystem/subcomponent. The fourth digit is not referenced [Nordmann 2000]. ATA chapter codes are generally divided into four categories, aircraft, airframe systems, propeller/rotor systems, and power plant systems (Table 4.1). By using ATA chapter codes, tasks can be grouped by component, and special focus can be applied to certain functional areas for further detailed analysis [NTSB 2005].

Table 4.1.1 ATA Chapters Code Summary

| ATA chapter code | Category | ATA chapter code | Category |
|---|---|---|---|
| | *Aircraft* | 54 | Nacelles/Pylons |
| 11 | Placards & Markings | 55 | Stabilizers |
| 12 | Servicing | 56 | Windows |
| 18 | Helicopter Vibration | 57 | Wings |
| | | 60 | Std. Practices-Props |
| | *Airframe systems* | | |
| 21 | Air Conditioning | | *Propeller/rotor systems* |
| 22 | Autopilot | 61 | Propellers/propulsors |
| 23 | Communications | 62 | Main rotor |
| 24 | Electric Power | 63 | Main rotoe drive |
| 25 | Equipment & Furnishing | 64 | Tail rotor |
| 26 | Fire Protection | 65 | Tail rotor drive |
| 27 | Flight Controls | 67 | Rotor flight control |
| 28 | Fuel | 70 | Standard Practices Engine |
| 29 | Hydraulic Power | | |
| 30 | Ice & Rain Protection | | *Powerplant system* |
| 31 | Instruments | 71 | Power Plant-General |
| 32 | Landing Gear | 72 | Turbine/turboprop Engine |
| 33 | Lights | 73 | Engine Fuel & Control |
| 34 | Navigation | 74 | Engine Ignition |
| 35 | Oxygen | 75 | Engine Air |
| 36 | Pneumatic | 76 | Engine Controls |
| 37 | Vacuum | 77 | Engine Indicating |
| 38 | Water/Waste | 78 | Engine Exhaust |
| 45 | Central Maintenance System | 79 | Engine Oil |
| 49 | Airborne Auxiliary Power | 80 | Engine Starting |
| 51 | Structures | 81 | Turbines |
| 52 | Doors | 82 | Engine Water Injection |
| 53 | Fuselage | 83 | Accessary Gear Boxes |

In addition to the service branch in the hierarchy, we can group all the activities by ship and use features we generated, then we have performance measures for each of the ships; certain ships with the same type can be grouped into a fleet, which represents a specific type of aircraft design, such as Boeing 737-300 and Boeing 777-200. At the fleet level, we can compare service performance between different aircraft types, which is a reflection of the design characteristics. If we treat the support and maintenance activities in an airline as a system, we can now investigate the characteristics of the system.

Furthermore, airlines with all their fleets integrated can be considered as a whole, such as Delta Airlines and American Airlines, to compare their service effectiveness at system of systems level.

Top-down approach is to decompose the system from top level down to subordinate levels. In the case of aircraft support and maintenance system, based on the two fundamental aspects of the system, we can decompose along two traces: From system effectiveness, to component effectiveness, to task effectiveness; and from system effectiveness, to fleet effectiveness, to ship effectiveness. System effectiveness in this context represents how effective a system is in the operation and maintenance point of view. Some metrics of effectiveness are availability, maintainability, reliability and cost.

Availability represents the degree to which a system suffers degradation or interruption in its service to the customer because of failures of one or more of its parts. In the content of aircraft support and maintenance, we use total service hours on a system to model availability of the system by assuming the system will be available to use when it is not in maintenance. Therefore, low total service hours means that a system has longer time in serving customer needs and higher availability.

Maintainability represents how easy to fix the problem when something goes wrong. Average service man hours per task are a sound metric to model maintainability. The service man hours per task is a composite concept that is a combination of how many hours a task takes and how many persons are involved. Therefore, it is not a simple time dimension, but a measure of the amount of effort on a task instead. Without other information, one can not tell the duration of a task, rather he will have an idea about how easy to perform the task, which is our intention. Typically, a system is assumed to have high maintainability, if it generally requires less service man hours in maintenance.

Reliability represents how stable a system is to remain in good working condition without the interruption of service. Service frequency is used to measure reliability. It describes how many times of certain services are carried out on a system in a given time period. The system is deemed less reliable if it is frequently serviced. If the system is less serviced, we say it is a more reliable system, assuming the problems are found in time, and the system is serviced when necessary.

Cost is always a crucial factor throughout the aircraft life cycle. Without the actual financial expense information of the maintenance services, we can measure the level of service cost by comparing the total service hours between tasks. Having labor cost to be the primary cost of the maintenance service, more service hours spent on a task would represent a more costly task. We generate a feature called CostRank, which ranks different tasks by the cost associated to them. All the tasks in a fleet are sorted in ascending order by the total service hours, and the task with the highest total service hours in a fleet has the CostRank of 1. Similarly, the task at the No. 10 position of the sorting result will have the CostRank of 10 in the fleet. By using the feature of CostRank, we can associate economic factors with the maintenance service at the absence of real financial figures, and concentrate on high cost ranked tasks that are one of the most important aspects of the services.

### 4.2.6 Visualization for Decision Making Support

The results from the mining need to be interpreted with easy-to-understand format, and visualization techniques can be applied to facilitate the presentation. The selection of visualization techniques will be based on the structure of the mining results, and the special needs of customers, and it can be some basic charting, or some advanced technique, such as multivariate scatterplot matrices.

Many statistical analyses involve only two variables: a predictor variable and a response variable. Such data are easy to visualize using various two-dimension plots. It's also possible to visualize tri-variate data with three-dimension scatter plots, or two-dimension scatter plots with a third variable encoded with, for example color. However, many datasets involve a larger number of variables, making direct visualization more difficult since human eyes are only good at low dimension (less than or equal to three-dimension).

Scatter plot matrices overcome the high dimension visualization dilemma by viewing slices through lower dimensional subspaces, which is one way to partially work around the limitation of two or three dimensions. Scatter plot

We can display an matrix of the multivariate scatterplots for each pair of features generated in previous steps, and arrange them in a specific manner for hierarchical analysis. Figure 4.2.3 shows an example of multivariate scatterplot matrix.

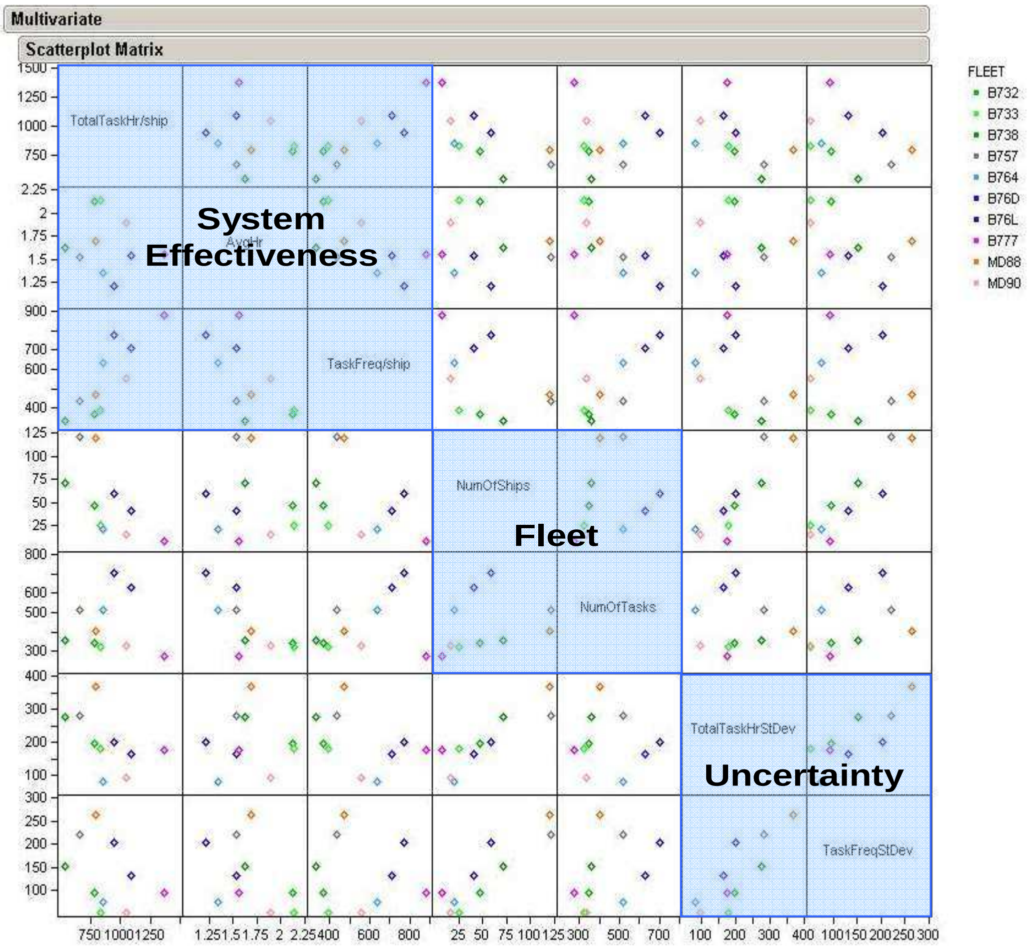



Figure 4.2.3 An Example of Multivariate Scatterplot Matrix

The previous discovered knowledge, together with the aid of visualization, can be transferred to the decision makers more easily, and equip them with deeper understanding of the problem, and resulting in better decisions.

Furthermore, as an iteration process, when users view the presented knowledge, more questions may be found requesting deeper investigation. The feedbacks from users can be sent to feature generation step for adjustment, and knowledge that is more relevant can be discovered to satisfy users' needs.

## 4.3 Feature Exploration

In this section, we will elaborate one of the most important steps in the knowledge discovery process – feature exploration. To most efficiently gain knowledge from a data set, which means optimize the amount of effort and maximize the amount of useful output, we need to identify a set of classification and evaluation criteria, called features. Features can be seen as the clues in the data. They could connect different things together, and could separate things from each other. An old Chinese idiom says: “Hua Long Dian Jing”, dotting the eyeballs brings the painted dragon to life, see Figure 4.3.1.

Figure 4.3.1 Dotting the Eyeballs Brings the Painted Dragon to Life

It emphasizes the importance of the eyes to the dragon. The eyes give the life to the dragon, and they are the spirit of the dragon. Features to a dataset are the eyes to a dragon, and features are the spirit of the data. Without features, a dataset is just a bunch of numbers gathered in a pool. With features identified, the data can be processes much easier, and analysis will provide more valuable and accurate information with less effort. Although critical, the methodologies on feature exploration in database are far from been established systematically under author's knowledge. Therefore, this research has a focus on the design of a methodology to explore the potential features in the given data and identify key features in the knowledge discovery process. We define three types of features as follows.

*Evaluation Feature*: An evaluation feature is a metric used to evaluate performance of a group of data. This type of features has only a key, and has no fixed value tied to it in the feature specification. The value of a key is quantity we need to get from the dataset in order to evaluate the certain group of data. For example, "Frequency of hurricane" is a feature to judge a region on probability of receiving potential damage from hurricanes, where "average number of hurricanes per year" is the key, and the value is a quantity we need to find out so that one can decide how safe the region is subject to hurricane.

*Specialty Feature*: A specialty feature is a special property of certain records, and it can be used to distinguish the records from others in a dataset. This type of features normally has a value associated with a key at a relation. For example, "maximum flight speed greater than mach one (V_Max > 1)" is a feature to identify supersonic aircraft from subsonic aircraft. The key is "V_Max", the value is the number "1", and the relation between the key and the value is the ">". As another example, "Having long ivory (Have_Long_Ivory = true)" is a feature to identify adult male elephants from adult female elephants, where the key is “Have_Long_Ivory”, the value is “true”, and the relation is “=”.

*Classification feature*: A classification feature is a simple field for the grouping purposes. This type of features has only a key, and has no fixed value tied to it in the feature specification. The value of a classification feature will be used to identify the groups. With a classification feature, the dataset can be divided into multiple groups. For example, “animal class” is a classification feature, which will divide the animals into natural groups, such as bird and fish. A classification feature is different with the specialty feature. The later one can only represent one group of records. However, a classification feature can be converted into a specialty feature by adding a value to it. For instance, “animal class = bird” is a specialty feature, which will identify all birds from other animals.

To aid the process of feature generation, we categorize features into different levels. Sometime a feature is obvious, and can be identified easily. For example, “flight Mach number greater than 1” is by definition the feature of a supersonic aircraft. In some other cases, feature identification requires domain knowledge. For instance, to identify “having long ivory” as a feature, one needs to have some kind of basic knowledge about elephant. Some other features require expert knowledge and significant amount of experience in that domain. For example, clinic diagnose sometimes is based on certain known features for a type of disease in doctor’s established knowledge. Next level of features is subtler, and can only be identified with some algorithm. Even for most experienced domain experts, it will be hard for them to spit it out simply from their mind.

As a sophisticated engineering process, the aircraft system life cycle activities may contain many features. Although some of them might be easily identified, the exploration of most of features will require the design of a special algorithm.

In this section, a hierarchical algorithm on feature exploration is presented. See Figure 4.3.2.

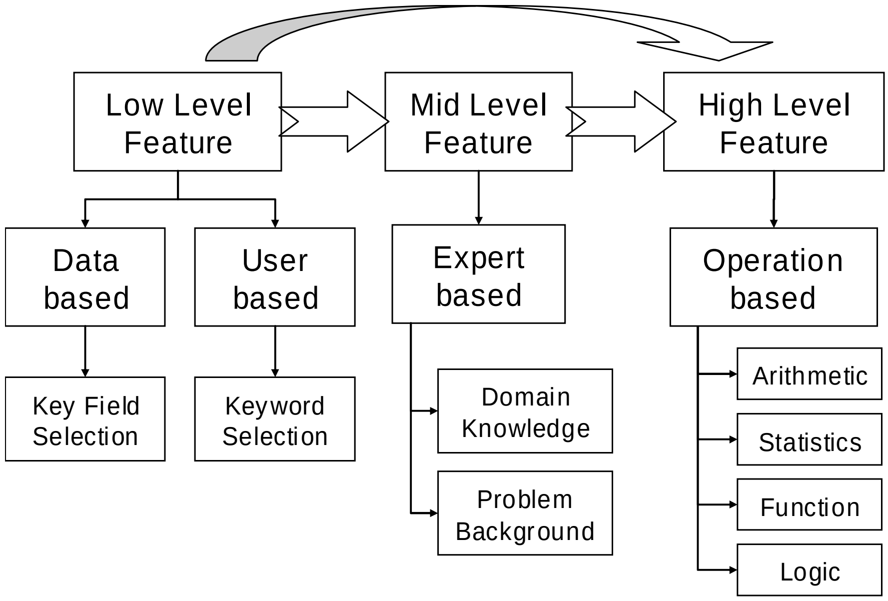



Figure 4.3.2 Hierarchical Feature Exploration

### 4.3.1 Potential Feature Generation

First step in the feature exploration is feature generation. Since features are closely related with the data and the specific domain we are investigating, they are not to be generated randomly out of nowhere. We need to find out various sources where feature can be created. Those sources are not limited to the data themselves in the given dataset, but are tied to the data. Based on the complexity of the feature generation and the source of the features, the features can be classified into different levels. From the lower level features, which are relatively straight forward in the cognition process and easy to be created with less knowledge and technique required; to the medium level features, which based on domain specific experience and expert knowledge; then to the high level features, which are primarily some mathematical abstractions. The classification of features is demonstrated in a pyramid structure as Figure 4.3.3. In this structure, lower level features are the fundamental of the higher level features, and the features in a higher level are created based on the features at lower levels.

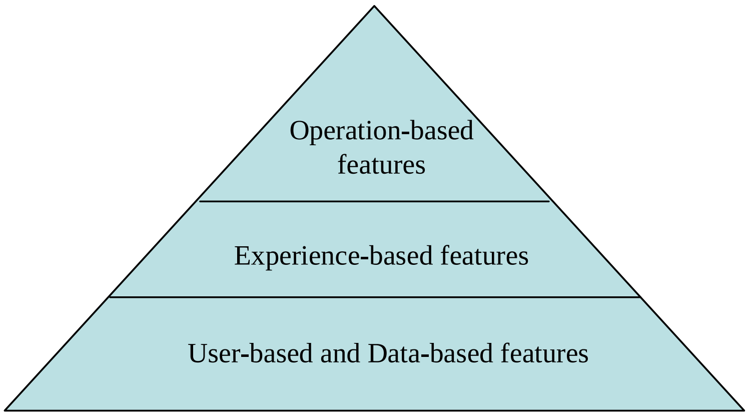



Figure 4.3.3 A Pyramid of Features

Then significance of each feature will be tested, and most influencing features will be selected in feature identification step.

4.3.1.1 Low-Level Feature

As a starting point, we begin to gather low-level features. Those features probably already exist in the data, and easy to be identified. There are several channels of sources, where low-level features can be extracted.

One is the user inputs, which will generate user-based features. If users have some basic knowledge of the problem itself or certain expectation of the problem, he or she can provide some thoughts to represent features. For example, if a user plans to explore the common design behaviors related with large commercial passenger transports with gross take-off weight greater than 20000 lb, several features can be defined. One apparent feature from this problem is “growth take-off weight > 20000 lb”, and another user-based feature could be “aircraft type = commercial”, and the third feature identified from the user can be “aircraft function = passenger transport”. Sometimes, the ideas from user inputs are not well defined, and need to be converted into keywords. Techniques, such as QFD, can be employed here to assist the process.

Another important source is the dataset itself, which will generate data structure based features. A dataset may contain hundreds and thousands of rows of records, and each field of a row represents a particular aspect of the record. Some fields in the dataset may be of special interest to some problem. For instance, the field “tail number” is particularly important if one is comparing the abnormal behaviors between individual aircraft in a dataset. Although all fields have their own purposes on representing the properties of a record, it is not efficient to make all the fields in the data as features since it may result in losing focus of the original problem by bringing in too much noise information. Therefore, we need to combine the field selection with the final goal of the problem, and use the defined goal to guide out feature generation process. With this selection process, some fields in the dataset can be selected as key fields, and enter the pool of potential features.

#### 4.3.1.2 Mid-level Feature

The medium level features represent some special properties of the problem domain. They are not general terminologies that one can obtain by common sense, and more specific towards the final goal of the problem. However, they are still some criteria that can be obtained directly, and they can be thought as a special group of fundamental features. The reason we separate them from the low level feature is to emphasize their special property. The generation of this level of features requires involvement from a special group of people, i.e. the domain experts, involved in the process. In addition, the features are expert-based features. The procurement of those features can stem from in-depth interactive communication with domain experts.

The features can be a result of specific *domain knowledge*. For example, in the aircraft maintenance domain, “man hours” is a term specifically related with the aircraft maintenance process.

They also can be more detailed knowledge related with *problem background*, such as certain problems usually faced in certain type of aircraft, which is called *fleet* in airline business.

4.3.1.3 High-Level Feature

Features at this level are not obvious, and not exist directly either in the data structure or in experts’ mind. It requires special designed algorithm and lots of computation to obtain them. However, on the other side, they are also more critical than other features at lower levels in knowledge discovery. Once generated, the high level features will greatly simplify the knowledge discovery process, and improve the significance of the result. High-level features are based on the low-level and mid-level features, and may be in the forms of combinations or derivatives of them. Some proposed approaches on creating potential high-level features are listed as follows:

*Arithmetic-based feature exploration*: Basic arithmetical operators can be used on lower level features to form new features, which will represent information abstraction at a higher level. Some operators, such as summation and average, when combined with grouping function, will be able to identify some characteristics of data in certain group. For instance, the “man-hour” is a mid-level feature, and it measures on how long a specific service job took on a specific aircraft in a specific fleet at a specific time and location. With previous identified lower level features, we can say man-hour (man-hr) is a function of multiple factors, such as task, tail number (tail-num), fleet type (flt-type), service date (s-date), and service station (s-stn). We define “task” as a unique type of job, such as replace noise wheel, which is identified by a unique task-id; the term “job” is referring to one individual service happened before; the term “fleet” represents a type of aircraft, such as Boeing777 or MD82.

*man-hr = f(task, tail-num, flt-type, s-date, s-stn)*

By grouping with task, we perform summation and average operations on man-hour for a certain task, and generate two high level evaluation features, i.e. total man-hour (*TotalManHourByTask*) and average man-hour (*AvgManHourByTask*). In the meanwhile, we can count the number of jobs in this tasks, which results in another evaluation feature, *FreqByTask*.

*FreqByTask* displays how many times the jobs in a particular task happen within the investigation period, and we can use this feature to find out if the aircraft is prone to a task or not. *AvgManHourByTask* reflects the mean duration of the jobs in the current task, and it can help us to see if a task is an easy task or taking lots of time. *TotalManHourByTask* represents how many maintenance hours were spent on a particular type of task, which is a combination of task happening frequency and duration, and is an overall measure of the behavior of the task. With those aggregated features, we are looking at the task level, which is a higher level than the individual jobs. We can see the overall characteristics for each of the tasks, and compare between tasks. Certain problematic tasks can be identified by those three features.

When we group records with tail number, and carry out summation and average operations on man-hour for a certain aircraft, and get two high level evaluation features, i.e. *TotalManHourByAircraft* and *AvgManHourByAircraft*. In addition, we can get *FreqByAircraft* by counting number of records in an aircraft. With those features, we are looking at the individual aircraft level, and all tasks are aggregated together. We can see the overall characteristics for each of the aircraft, and compare between aircraft. Certain aircraft having longer maintenance hours or liable to hard-to-fix problems can be identified by those two features.

If we use summation, average and count operations with grouping on fleet type, we can get more abstract features regarding each type of the aircraft. For example, *FreqByFleet*, *TotalManHourByFleet* and *AvgManHourByFleet* are fleet level features related to task length and frequency. Where different fleets can be compared side by side, troubles fleet can be easily identified, and problems within the fleets can be investigated further.

*Statistics-based feature exploration*: Statistical formulas can also help to generate high level features out of lower level features. For instance, accumulative functions, such as probabilistic distribution function and cumulative density function, calculate individual and cumulative frequencies for a range of data. It generates data for the number of occurrences of a value in a data set. The following example will explain how it will work.

We can create new features by using frequency distribution and combining with the high level features discussed above. One example is to combine *AvgManHourByTask* with cumulative frequency for each of the fleets. The idea is to divide *AvgManHourByTask* of all tasks in a fleet into hourly buckets, and count the occurrence of the tasks in each of the buckets. Thus, we have a new feature, *Freq_TaskHourBucketInFleet*, which will show the number of tasks in each of the *AvgTaskHours* bucket in a fleet. Cumulative frequency is calculated by summing all the occurrences where the *AvgManHourByTask* is less than the current bucket value, and greater than the lower bucket value. For example, we can divide the *AvgManHourByTask* into hourly buckets, i.e. 0-1, 1-2, 2-3, 3-4, etc. A task with *AvgManHourByTask* =1.5 will fall in 1-2 bucket. Since the number of tasks with average man hour greater than five is not as many as that with shorter hours, we can merge longer hour buckets, and have 5-10 and 10+, thus we have seven buckets to fill in. With this *Freq_TaskHourBucketInFleet* feature, we can identify the general task trends for a certain fleet. When apply cumulative function on this feature, we can get another new feature *CumulativeFreq_TaskHourBucketInFleet*, which will count all the occurrence of the tasks whose average man hours are less than a bucket value. The knowledge we could get from those feature are: Is the fleet easy to maintain or not? Does it usually take longer time to fix an aircraft in this fleet? What percentage of the service takes longer time in this fleet? And so on.

Moreover, instead of *frequency*, we can use some other high level features as merits to uncover more hidden knowledge in *AvgManHourByTask*. For example, applying *TotalManHoursByTask* on *AvgManHourByTask*, we will have new features as follows:

*TotalTaskManHours_TaskHourBucketInFleet,* which is a representation on the total maintenance hours spent on a specific set of tasks, whose average maintenance hour falls in a specific bucket for a fleet. For example, for the bucket 2-3 in Boeing777, *TotalTaskManHours_TaskHourBucketInFleet*=700 means total 700 hours was spent on all the tasks, whose average duration is between 2 hours and 3 hours. With this feature, we can compare different buckets in a fleet, or same bucket between fleets. In this feature, the total man hours of tasks are summed up in the buckets, instead of just counting heads, and the fleet property will be focused on the total maintenance time spent in each bucket, not total number of tasks in each bucket.

*CumulativeTotalTaskManHours_TaskHourBucketInFleet* is the accumulated total maintenance hours for all the tasks whose average man hour is in the current bucket or in the lower buckets.

Taking it one step further, we can replace the base feature, *AvgManHourByTask*, with other features, for example, *TaskFreq*, Thus we will have new features as follows:

*Freq_TaskFreqInFleet*

*CumulativeFreq_TaskFreqInFleet*

Replacing *frequency* with *TotalTaskManHours*, we have

*TotalTaskManHours_TaskFreqInFleet*

*CumulativeTotalTaskManHours_TaskFreqInFleet*

As one can see, the features are getting more complex, and they become more abstract, and the knowledge is closer to the surface for us to grab.

*Function-based feature exploration*: If the distribution of data can be assumed as certain distribution functions, such as Normal, Weibull, or Exponential, one may be able to create features with functions and their parameters. For example, if we can assume normal distribution to be the distribution of the task man hour of a particular task, we can use mean (μ) and standard deviation (σ) of the normal distribution as the features of this task. The probability density function of the normal distribution:

$$f(T) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

If we can assume Weibull distribution to the service time interval of identical tasks, we can use shape (β), scale (η) and location (γ) parameter as features. The probability density function of the Weibull distribution is:

$$f(T) = \frac{\beta}{\eta}\left(\frac{T-\gamma}{\eta}\right)^{\beta-1} e^{-\left(\frac{T-\gamma}{\eta}\right)^{\beta}}$$

With these three features, we can calculate the mean ($\overline{T}$) of the Weibull PDF, which is also called MTTF (Mean Time To Failure, used for unrepairable system) or MTBF (Mean Time Between Failure, used on repairable system).

$$\overline{T} = \gamma + \eta \cdot \Gamma\left(\frac{1}{\beta} + 1\right)$$

Where $\Gamma(\frac{1}{\beta}+1)$ is the gamma function evaluated at the value of $(\frac{1}{\beta}+1)$

The gamma function is defined as,

$$\Gamma(n) = \int_0^\infty e^{-x} x^{n-1} dx$$

The reliability function of Weibull function is,

$$R(T) = e^{-\left(\frac{T-\gamma}{\eta}\right)^\beta}$$

Which is one minus the cumulative density function.

*Logics-based exploration*: Some logic process, such as if-then-else condition, can also aid us in generating high level features. Some examples of logic-based feature exploration are shown below for a maintenance database investigation.

Since there are too many tasks listed in the database, we may set our focus on important tasks, which could be in one of the following two categories:

1. Tasks take long time. Some tasks usually do not appear in the daily services. However, it takes a great effort to finish when it happens. This type of tasks could post a big impact on regular flight schedules. Hence, those tasks are critical tasks, and needed to take a closer look. We use the following logic expression as a feature to facilitate our investigation.

$AvgHoursByTask > \alpha$

2. Tasks happen very often. Although individual task may not take too long, it has become a bigger concern if it occurs in a high frequency. It naturally leads us to a new feature.

$TaskFreq > \beta$

However, this feature could mislead us since the fleets have different number of ships. Therefore, we redefine the feature to a new formula that divides the frequency by the number of ships in each fleet, and removes the fleet size issue.

$TaskFreq/Ship > \beta$

Although we have the above features, one question may arise: What are the values of $\alpha$ and $\beta$? This is a topic to be discussed in the next section.

### 4.3.2 Feature Identification

Feature generation is to generate features as many as possible. With the guidance of the defined goal, it tries to cover every aspect of the data set from domain knowledge, expert experience, data set structure, possible mathematical derivatives, statistical characteristics, logical properties, and so on. However, not all the features are suitable to describe the specific data set. Some features might be overlapping, and some features might not be available in the data, and some features are simply not important in the specific domain. Moreover, we also need to refine features to make them most significant. In another word, we need an algorithm to find out the values of the important features that can be applied to the give data set and create significant classification effects on it.

The following example is to illustrate the process of feature identification.

Assume that we have an evaluation feature E1, a classification feature C1, and two specialty features S1>$\alpha$, and S2>$\beta$. We will use the following algorithm to determine the value of $\alpha$ and $\beta$

As for the variables in the features, such as $\alpha$ and $\beta$ mentioned in feature generation, we need an algorithm to decide the best values for them. If we are using the features to seek abnormality in the data set, we can look at classification and find out which class of data represents significant difference to other classes. One of the performance measuring criteria is the significance of classification effects on the given data set. We could use the standard deviation, which is a measure of how widely values are dispersed from the average value (the mean). However, it may be moderate if there are only a small portion of outliers. Since we are seeking outliers, we use Range/Mean to evaluate the performance of a feature, where Range is Max-Min.

The pseudo code for the algorithm is as follows.

```
For α from 0 to 5 increment by 1
  For β from 0 to 5 increment by 1

    Select
      Sum(E1) AS TotalOfE1,
    From table1
    Where
       S1 > α
       and S2 > β
    Group by C1

   maxE1=max(TotalOfE1)
   minE1=min(TotalOfE1)
   avgE1=average(TotalOfE1)
   evaE1=(maxE1-minE1)/avgE1

   store(α, β, evaE1) into array-A

  End for
End for

Select max(evaE1) from array-A
and set α*=α, β*=β
```

From above algorithm, we will have a set of $\alpha$ and $\beta$, where $\alpha^*$ and $\beta^*$ are corresponding to the data with biggest gap between fleets on *E1*.

Once we have a pool of features, from every source we explored, we can start to apply those features on the data set, and find out what knowledge will be extracted from each of features.

## 4.4 Knowledge Management for Aircraft Life cycle Design Decision Support

After knowledge has been distilled from the existing data, it can be presented to decision makers directly with the knowledge presentation module. Meanwhile, it should also be kept in a good place for future reference, and efficiently managed so that it can be accessed easily by users no matter when and where. Because of the complexity of the aerospace system, the amount of knowledge and information are immense even in each of the subsystem of the whole system, and the knowledge is in different formats. When combining all the phases in the aircraft life cycle, the volume and heterogeneity make manually managing the knowledge impractical. Therefore, it is important to create an organized structure to store the knowledge in a systematic manner, so that the knowledge can be easily accessed by decision makers.

### 4.4.1 Analysis of Requirements

To create such a managing environment to efficiently handle the knowledge in aircraft cycle, certain characteristics are required (Figure 4.4.1):

*System structure*: The most important requirement on the environment is the overall structure of the stored knowledge. The knowledge should be organized in systematic manner following the aircraft system's hierarchy, and layers of abstraction should be used to accommodate the users' focus level. Thus the whole structure can be easily browsed based on the level of information requested.

*Centralized storage*: Knowledge should be stored at a centralized location, and managed together. The advantages of the centralized storage are easy management and maintenance, consistency, and easy access control. The main disadvantage is the storage size that may be an issue if too much information is gathered. As an alternative, we can keep only the reference to the remote content. It sounds like a good idea, since it will reduce the storage requirement, and release the management overhead. However, it creates heavy dependency on the remote sites. The remote resource may not be always available overtime, such as the service may be down for maintenance, the remote site may change its structure and reorganize the content, or simply no longer keep the content, then the reference is not valid any more, and the knowledge is lost when this happens. Network connection is another issue, for the content has to be transferred via a network, which is relying on the connection stability. Access control is harder to achieve, because the control is primarily in the hand of remote sites. Based on the above comparison, a centralized storage is preferred to maintain overall completeness.

*Easy to search*: Whenever something is stored, it is essential to make it searchable. Especially in an aircraft system that has huge volumes of knowledge, the capability to search can never be over-stressed. The search can be done in multiple ways. Among them, the most important mean is by keywords, where users can seek related knowledge with the central ideas that the knowledge is about. Other clues, such as author and date, may be used to enhance the searchability.

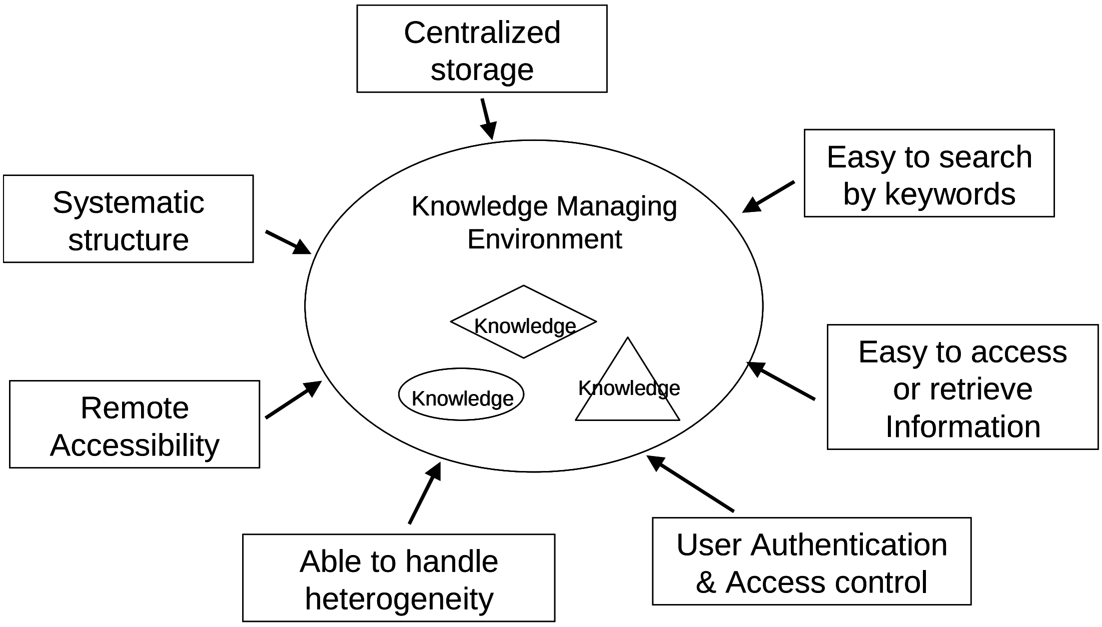



Figure 4.4.1 Requirement of Efficient Knowledge Management

*Capable to handle heterogeneity*: The knowledge can be in many formats, such as document, picture, audio, video, software program, database, or a simple spreadsheet. The environment needs to be able to accommodate all possible digital media formats. Other media that are not digital can be converted to digital formats for storage. Researches have been done on the traditional media to digital media conversion process, which is not in the scope of this thesis.

*Easy to access or retrieve*: Once users have found the related knowledge, the environment needs to have a user-friendly interface, present all the information related to the knowledge, and provide methods to access the knowledge itself. It means the knowledge needs to be downloaded to a user's computer. It is preferred that the environment is able to guide the user's computer to find such an appropriate program locally to view the knowledge.

*User Authentication and access control*: Since knowledge sometimes contains proprietary content, the environment needs to provide a mechanism to manage content access so that certain knowledge is only available to a specific group of users. In addition, the overall access to the environment should be protected by user name with password.

*Remote accessibility*: In the information era, the usage of an environment will be significantly limited if it can only be accessed locally. The knowledge stored in the proposed environment should be accessible by users from remote sites via the internet or an intranet, no matter where they are located.

Based on the previous requirements, a framework of the knowledge management environment is developed.

### 4.4.2 Framework for the Knowledge Management Environment

*Structure Definition*: First, we define a knowledge structure based on the hierarchy of the aircraft system. The structure is decomposed according to the phases of the aircraft life cycle. Therefore, we will have a top level component for each of the aircraft life phases. Such as Design phase, Production phase, and Operation phase. In the design component, all the knowledge primarily associated with design is stored, and a crosslink can be created to other components if there is information involving other phases.

Top level components are further divided into sublevels to accommodate subsystems in each stage of the aircraft life. For example, a design component can be divided into conceptual design, preliminary design, and detailed design. A preliminary design can be divided into aerodynamics, structures, propulsion, and so on.

The framework is flexible and extensible. The defined structure can be refined over time to best fit for new available knowledge.

*Knowledge Entering Procedure*: With the defined structure, we define the procedure by which the knowledge is entered into the system. This is based on the abovementioned characteristic requirements, to better manage the knowledge and make the knowledge accessible to the users.

To enable keyword search, we require the keywords to be given for each of the knowledge entered. In addition to the keywords, a brief abstract of the knowledge is preferred, so that the content of the knowledge is better defined and summarized for future search and browse. The knowledge should contain one or more media files that present the knowledge itself. Each of the media files is companied with a brief description for the content of the file. A field to specify the format of the media file is also required, so that the environment can guide the remote computer to find a proper program to open the file. Some other information, such as author, date, and comments, can also be entered with the knowledge for advanced searches. When all the information is defined, the media files together with the description are uploaded to the central location for storage and remote accessing.

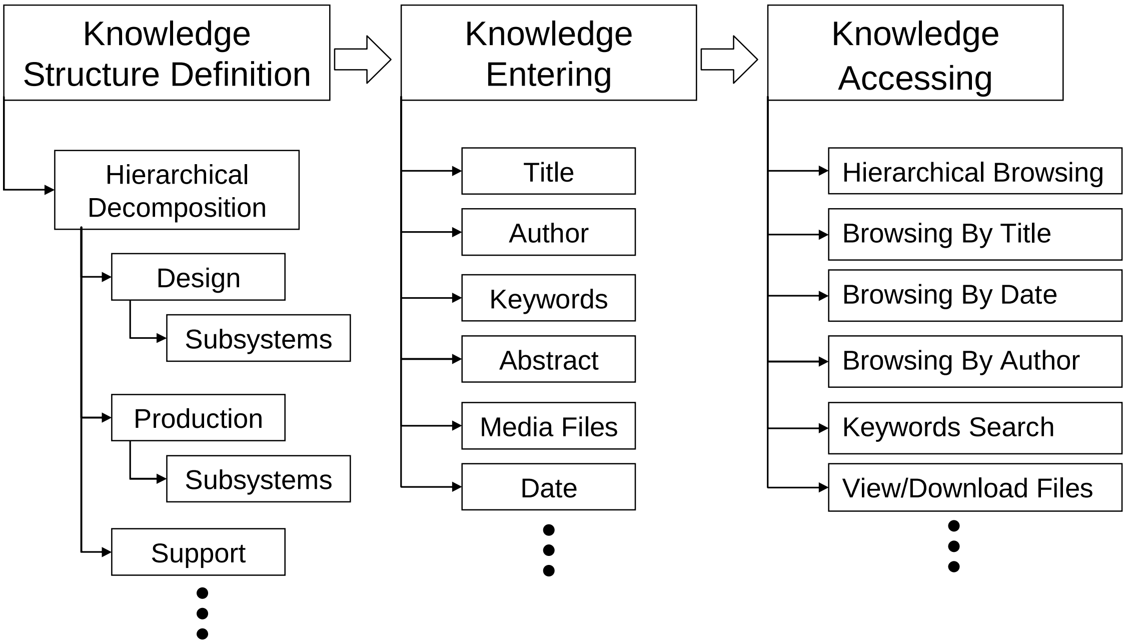



Figure 4.4.2 Knowledge Management Framework

*Knowledge Accessing*: The purpose of store and manage knowledge is to access it when needed. The process of access knowledge is defined as first finding the related information, then viewing the description, and finally getting the knowledge and information detail. To fine the relevant knowledge, we provide browsing and search capabilities. A user can browse by hierarchy and scan all available knowledge stored in a certain level, if they are not sure what to find. If the user has a clear goal on what to look for, he or she can specify the keywords and search for related knowledge. Each of the items found can be expanded to display a detailed description. Once the user decided to view the detail, the knowledge can be downloaded to local computer for further review.

### 4.4.3 Analysis of Available Alternatives

There are a few existing digital repository systems available, and we evaluate them, and pick one as the foundation of our framework. For prototyping, we select one based on our requirements, and customize it to meet our special goal. Some of the systems are briefly described below.

EMC Documentum [Documentum 2005] is a commercial software that helps user create content with common desktop applications and easy-to-use content authoring templates. Content – or unstructured information – includes digital text documents, engineering drawings, XML, still images, audio and video files, and many others. It provides a content service to create, manage, deliver, and archive all types of content from text documents and spreadsheets to digital images, HTML, and XML components. It assists the process services with the capacity to define, model, manage, and analyze business processes consistently and reliably across multiple organizations, systems, and applications. It also has a repository service to reduce risk and protect information assets with built-in security, a compliance infrastructure, and storage optimization.

DSpace [DSpace 2005] is an open-source digital library system to capture, store, index, preserve, and redistribute research material in digital formats. Jointly developed by MIT Libraries and Hewlett-Packard Labs, the DSpace software platform serves a variety of digital archiving needs, such as institutional repositories, learning object repositories, e-theses, electronic records management, digital preservation, publishing, and more. DSpace integrates a user community orientation into the system's structure. This design supports the participation of the schools, departments, research centers, and other units typical of a large research institution. As the requirements of these communities might vary, DSpace allows the workflow and other policy - related aspects of the system to be customized to serve the content, authorization, and intellectual property issues of each. DSpace is also focused on the problem of long - term preservation of deposited research material. Some of the system's adopters are actively engaged in research and development in this area. Over time, this should allow DSpace adopters to offer services both for hosting institutional repository content and maintaining the content for archival periods.

Fedora [Fedora 2005] is an open-source general purpose repository system developed jointly by Cornell University Information Science and the University of Virginia Library. Based on the Flexible Extensible Digital Object and Repository Architecture (Fedora), the system gives organizations flexible tools for managing and delivering their digital content. Its digital model supports multiple views of each digital object, and each digital object can be a locally-managed content or make reference to remote content. Fedora provides an open-source repository software and related services to serve as the foundation for many types of information management systems.

Our goal is to create an affordable aircraft life cycle knowledge management environment as an open system available to students and academic researches. Therefore, we will use a free foundation. DSpace and Fedora meet this requirement. Among those two alternatives, DSpace provides some extra preferred functionalities:

- Limit Access at File/Object Level for refined access control.
- Full text search to allow users to search all defined descriptive fields.
- Email notification for submitters and content administrators
- Personalized system access for registered users
- Metadata review support to verify the correctness of the description entered.

Based on above analysis, the DSpace system provides a convenient approach to manage data, information, and knowledge with extra capabilities. Therefore, we use

DSpace as the testing environment of the knowledge management for the aircraft life cycle decision support.

## 4.5 Knowledge Integration with Data Configuration Control

As a special category of the knowledge management, in this section, we deal with the integrity of knowledge. Integrity is "an unreduced or unbroken completeness or totality." (WorldNet 2003). Knowledge integration in the context of this thesis is to achieve the completeness of the knowledge, which requires a linkage between the knowledge and the information where knowledge is rooted from. More specifically, we aim at creating a complete knowledge set for surrogate models.

### 4.5.1 Motivation

Most engineering design problems require expensive simulation and optimization, which are sometimes associated with thousands or even millions of evaluations. Surrogate modeling is an effective and popular way to alleviate the computational cost by approximating the actual function with constructed algebraic expression

Surrogate models are widely used in the aerospace industry and other areas for modeling, design and optimization (Glaz 2006, Goel 2006, Choi 2005, Kodiyalam 2004, Ong 2003, Nair 2002, Otto 1996). Surrogate modeling includes some techniques to create approximate models, such as response surface modeling (Mavris 1997), Kriging (Krishnamurty 2004), evolutionary algorithm modeling (Ray 2006), neural network modeling (Biltgen 2006) and Bayesian Gaussian Process (Nair 2001), of the physics-based engineering programs or actual experiments. When properly constructed, the surrogate models will maintain the fidelity of the proprietary programs. There are many advantages of the surrogate models:

They are fast-running and time-saving, comparing to those large-scale integrated physics-based simulation programs which are extremely costly and complicated. Multidisciplinary optimizations and robust designs with large number of iterations can be quickly performed with those simplified models if their accuracy is sufficient. With the support of surrogate models, one can rapidly assess the alternative concepts with minimal time and monetary expenditures.

Intellectual property resides in the complicated programs is protected since simplified models are only problem specific equations. The process of creating the results is encapsulated in the equations and can not be reverse engineered. The collaboration between partners is made safer.

Surrogate models are easy to run and integrate to other programs across platforms since they are simple mathematic equations. The collaboration between partners is made easier.

However, there are some drawbacks of the surrogate models: the accuracy to represent complicated physics-based programs with simple models is always an issue. It can be resolved by leveraging on the respective disciplinary experts, and selecting proper modeling methods. The methodologies to improve the fidelity of the approximation are out of the scope of this thesis.

There is significant information loss in the creation process of the surrogate models. With the current practice of model generation, the only focus is the surrogate models themselves, which is acceptable if we just want the equation and never look back.

However, it is not the case when something has changed and we want a new model. For example, the range of an input variable may shift, and the distribution of a variable may change. Although it could be a slight change, all the cases need to be rerun, and the surrogate model needs to be recreated. The problem in it has two aspects: All the case data were lost, and we have to rerun everything from scratch; the configuration of the model is lost, such as what model effects are significant and need to be constructed and tracked. Great amount of effort is needed to produce a proper model configuration for an accurate model. In another word, the knowledge contained in the surrogate models is missing its integrity.

The creation of a surrogate model typically involves an expert team to work together for simulation and modeling, which is expensive, and sometimes it is simply not affordable to reassemble the team and create new models when conditions are changed. In an engineering field, some models are similar regarding to tools, goals and fidelity. It is a waste of resource, and sometime cost prohibiting, to repeat the process for slightly different models. If we can preserve the case data created and keep track of the model creation, we only need to run a few additional cases, and be able to recreate a new model with the exact sequence and parameters of the previous model. With much less cost running cases and model configuration, reusability is highly desired.

The traditional model approach neither tracking the creation process, nor keeping the valuable case data systematically, and the data with the configuration of the model are lost. Therefore, models are not reusable, and the previous case data are not available for new model generation.

A systematic approach is important to increase the model reusability by extracting and retaining data configuration in model creation process, and thus establish knowledge integrity.

### 4.5.2 Approach

Based on the motivation discussed in the previous section, one examines the general characteristics of the modeling process for the purpose of maintaining knowledge integrity, and propose a systematic approach to capture data configuration information in every step of the modeling process.

The typical modeling process includes the following steps:

Define the problem. It is to clarify the goal of the problem, and map the subjective and qualitative requirements into some engineering, or mathematically quantifiable metrics to measure the success of the system. There is a need at this point to create a mapping between the “voice of customer” and the “voice of engineer”. A tool called Quality Function Deployment (QFD) can be used for this purpose (Crow 1996). A sample of QFD matrix, or “House of Quality”, to identify customer needs is illustrated in Figure 4.5.1.

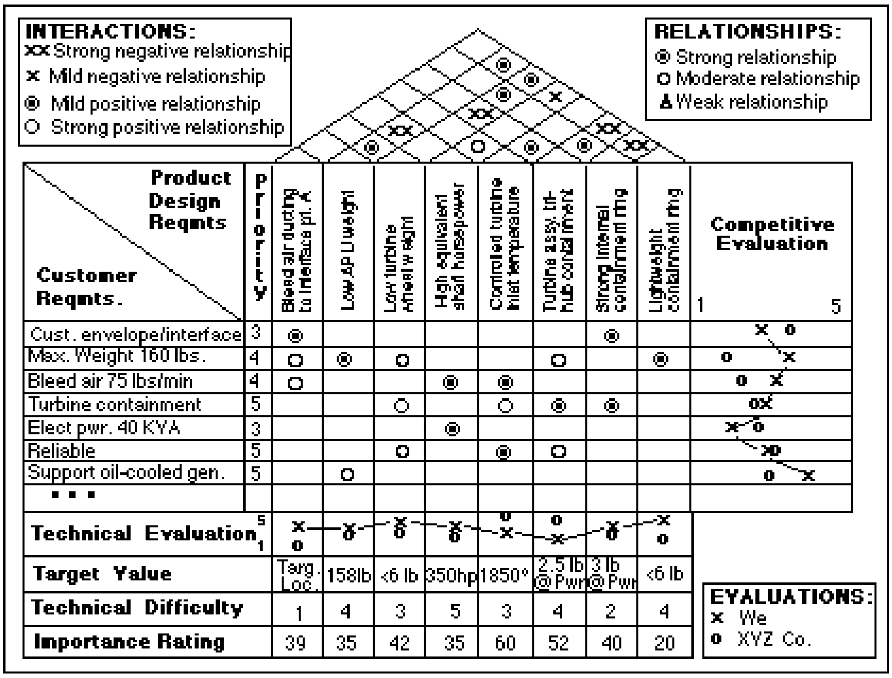



Figure 4.5.1 An Example of a QFD matrix

Select the modeling and simulation environment. Once the problem goal is defined, a set of modeling and simulation programs are selected to perform the analysis. In this step, some criteria of the codes, such as fidelity, cost, running time, compatibility, availability, and etc, are evaluated and compared based on the final goal of the problem.

Define the design space. The team first defines the technology and concept space, and creates a baseline from the alternative concept space, which can be done via the Morphological Matrix (an example is shown in Figure 4.5.2) or through brainstorming sessions. Primary product attributes include the physical design parameters that describe a characteristic of the system.

| Functional Decomposition | | Alternative 1 | Alternative 2 | Alternative 3 | Alternative 4 |
|---|---|---|---|---|---|
| config | Vehicle | Conventional Wing & Tail | Wing, Tail & Canard | Flying Wing | |
| | Fuselage | Cylindrical | Double-Bubble | Oval | |
| Mission | Range (nmi) | 5000 | 6000 | 6500 | |
| | Passengers | 350 | 450 | 550 | |
| | Mach Number | 0.75 | 0.8 | 0.85 | |
| Propulsion | Engine Type | Turbine Bypass Engine (TBE) | Variable Cycle Engine (VCE) | Fan-on-blade (Flade) | Mixed Flow Turbofan (MFTF) |
| | Materials | Conventional | Composite | | |
| | Combustor | Conventional | RQL | LPP | |
| | Nozzle | Conventional | Internal Flow Alternation | Mixed Ejector | |
| | Secondary Power | Auxiliary Power Units (APU) | Fuel Cell (FC) | Hybrid | |
| Aero | High Lift Devices | Conventional Flaps | Conventional Flaps & Slots | Circulation control (CC) | |
| Struct | Materials | Aluminum | Titanium | Composite | |
| | Process | Integrally Stiffened | Spanwise Stiffened | Monocoque | Hybrid |

Figure 4.5.2 A Sample Morphological Matrix for Military Aircraft

Where, RQL stands for Rich-burn/Quick-mix/Lean-burn Combustor, and LPP stands for Lean Premixed-Prevaporized combustor.

In conceptual and preliminary aircraft design phase, all of the design parameters should not be fixed but should vary within some specified range, and leave flexibility for future changes. This is to accommodate the inherent uncertainty in the design process. Within the context of surrogate model creation, the key design variables (with associated ranges) define the design space of interest for a given alternative concept. These design variables are often referred to as "control" factors, or variables that are within the designer's control. These key design variables, and associated ranges, define the design space for system feasibility exploration. The design variable ranges are chosen such that the largest possible deviations in the given baseline configuration may be captured.

Define design of experiment (DoE) is a technique to study the interactions between the design variables and their effects on the response metrics. Different designs can be used to create a DoE table, such as Full Factorial design, Central Composite design, Latin Hypercube design, etc. Even with the same design type, the number of running cases could vary. Each of the design has its advantages and drawbacks, and the selection is typically based on the fidelity requirement and time and budget constraints.

Once the design of experiment is defined, the actual analysis tools need be run based on the DoE configuration to generate a set of cases for surrogate model creation. This may be a lengthy process since it involves multiple runs of the simulation programs, which can take hours even days to finish. The proper selection of tools and DoE design will result the success and efficiency of this step.

Different surrogate models can be selected based on the designer's choice, such as response surface models, Kriging, evolutionary algorithm models, neural network models, Bayesian Gaussian process models, and so on. Each modeling approach has its parameters, and good combination of those parameters will result in good models.

With the properly configured parameters, surrogate models are created as a set of equations, which approximate the interaction between variables and responses. This step can be done with the help of mathematics or statistics software, such as MATLAB and JMP.

Once the model is created, the accuracy of the model needs to be checked to insure a good approximation, which includes not only the design points, but also the off-design points. If the results are not as good as expect, one can exclude some case if reasonable, or have some sort of model transformation, or some iterations of the entire model creation process need to be done for accurate models.

In this thesis, a standard process, mapping each of the modeling process, is created to ease and regulate the information capture process (Figure 4.5.3).

First, the different model creation processes are examined for commonality. Then the standard template is defined based on the commonality. The template can be applied to gather the modeling process information along with the progress of the modeling, and the content in the template can be updated while the model is updated. When the modeling process is finished, the extraction process is completed as well, and the information can be achieved in a model library for future reference. With this information the extraction process, data configuration control is captured, and the reusability of models is improved since the reference is kept with the models.

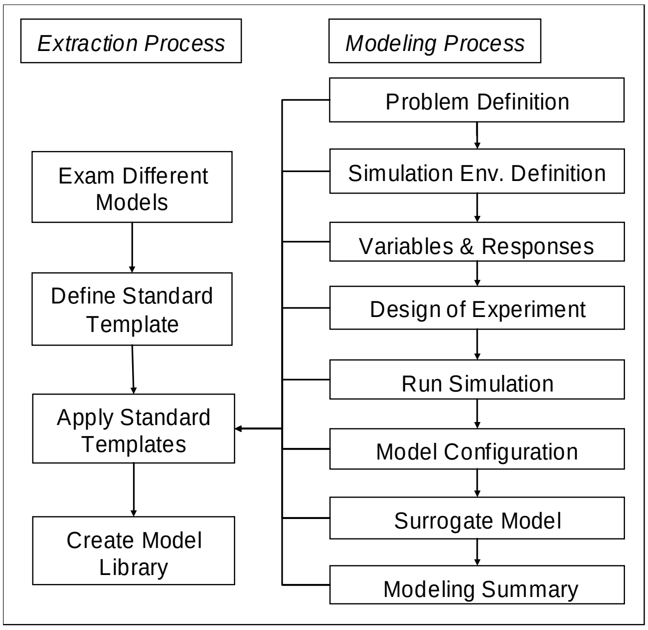



Figure 4.5.3 Process of Data Configuration Control

A sample of the data structure for data configuration control is illustrated with a mind mapping format in Figure 4.5.4. Centered with the surrogate model, we have various types of related information, which records the process and supports the creation of the model. In the model definition branch, we have model description, which clarifies the scope of the model and the goal of the problem, and the number of inputs and outputs. In the modeling tools branch, we capture the simulation environment by recording the information of all the tools needed in the environment. The set of the tools is matched by a collection of records. For each modeling tool, the template records the description of the tool, version of the tool, author information, running platform, language and user manuals is available. This is particularly useful when the actual simulation tools are constantly updated. Even the tools are modified, one can still track down the corresponding version of the tools which create the surrogate model, and be able to rerun the correct tools when needed. The variable summary branch keeps the information on all the design variables, such as description, baseline, ranges, unit, format, and distribution of the variables. The response branch is similar to the variable branch, which records the information about the responses, including description, format and units. The DoE branch maintains the information about the creation of the DoE table, such as design type, number factors, number of runs, and the actual DoE table. The model specification branch accommodates different type of models, such as Response Surface Models and Neural Network Models. For a Neural Network model, it records number of hidden nodes, number of tours, maximum number of iteration, overfit penalty and converge criterion. For Response Surface model, it records the set of response variables, model effects and emphasis of the model. In the model summary branch, the template records all pertinent model parameters. This encompasses the approximation equation with the standard error of each term, the accuracy of the model, which includes the R square, and some plots for visualization, such as actual by predicted plots and residual by predicted plots, while a prediction profiler can also be included for user interaction with the model.

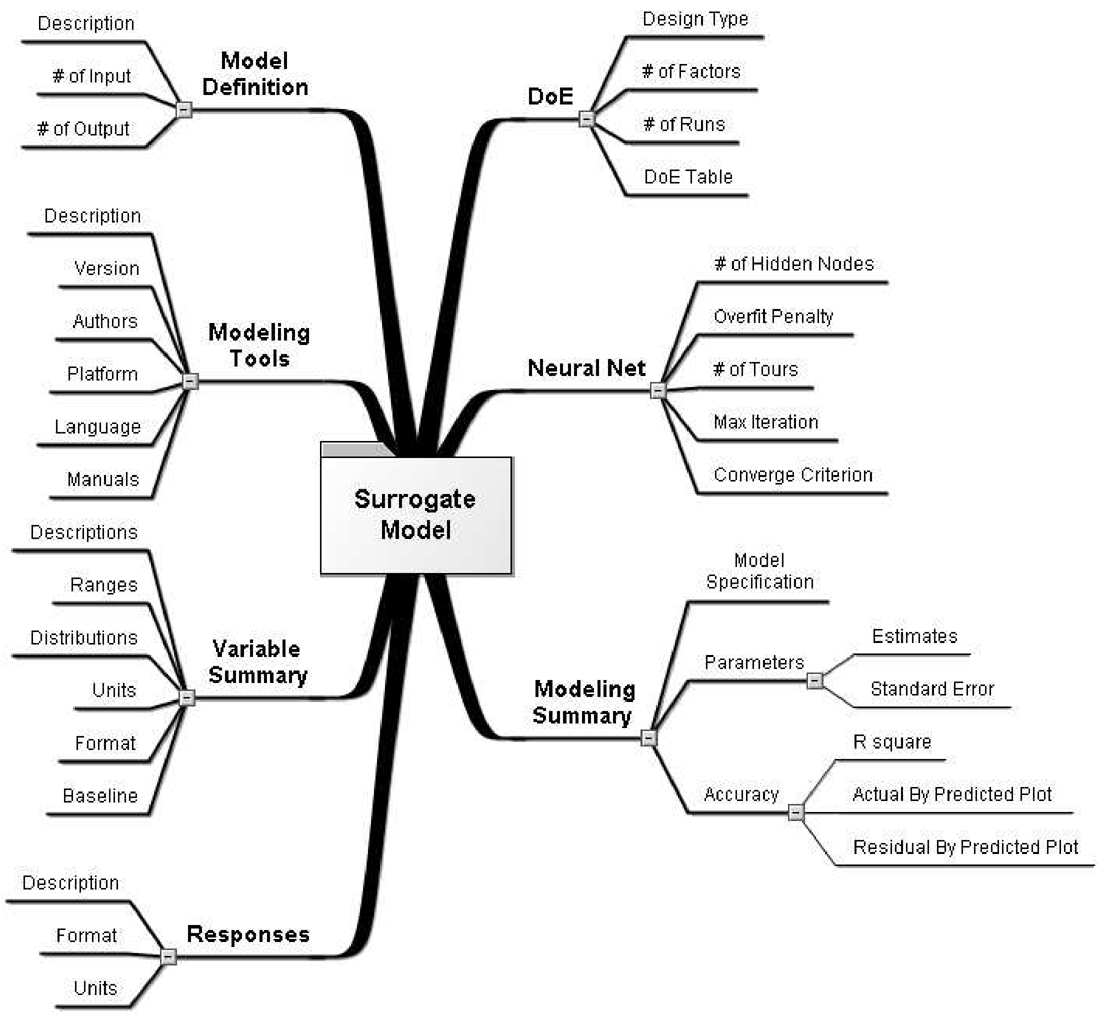



Figure 4.5.4 A Sample Data Structure of Data Configuration Control

Based on the data structure, one can create templates to take valuable information in parallel with the model generation process. There are two major phases in the model process, i.e. data preparation phase and model generation phase. The data preparation phase includes problem definition, simulation environment selection, and variable and response selection. Because of the simplicity and popularity of available software, we use Microsoft Excel to store information in this phase. A sample set of the template illustrates the concept with a sample problem and sample settings in Figure 4.5.5 and 4.5.6. Since we provide a standard template, it is very flexible, and the user can append any number of tools, variables and responses, and the columns can also be expanded to accommodate special needs. The user is free to copy or enter the related information and update it when needed.

| | **Model Definition** | |
|---|---|---|
| Goal | a goal for sample problem | |
| | | |
| Number of Input | n1 | |
| Number of Output | n2 | |
| | | |
| | **Analysis Tools** | |
| | | |
| Name | Tool 1 | Tool 2 |
| Description | sample tool 1 | sample tool 2 |
| Version | 1.0 | 5.4 |
| Operating System | windows | all |
| Language | English | English |
| Company | company A | company B |
| Last Modified | 1/1/2000 | 5/20/2006 |
| Modified By | author AA | author BB |
| Contact Phone | 203-123-3456 | 404-894-1111 |
| Contact email | aa@a.com | bb@b.com |
| Manual file | tool1.pdf | tool2.doc |
| Sample file | sample1.txt | |

Figure 4.5.5 Sample Model Definition and Analysis Tools

| Input Variables | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Varied | Name | Description | Location | Format | Unit | Baseline | Lower Limit | Upper Limit | Distribution | Ref |
| Yes | X1 | a sample variable | filename1 | integer | inch | 100 | 95 | 105 | uniform | |
| … | | | | | | | | | | |

| Responses | | | | |
|---|---|---|---|---|
| Name | Description | Location | Format | Unit |
| Y1 | a sample response | filename2 | float | lb |
| … | | | | |

Figure 4.5.6 Sample Template for Input Variables and Responses

The model generation phase includes DoE table generation, Model parameter configuration, model generation, and model verification. Many people use a statistics and graphics tool from SAS Institute, called JMP, to create their surrogate models. JMP has a strong interaction and support capability. The journal feature in JMP is used to record the modeling process. A template in a JMP journal file is created with place holders as displayed in Figure 4.5.7, such as DoE Design, DoE Table & Responses, Model Specification, and Model Summary, and user can insert information in each of sections accordingly.

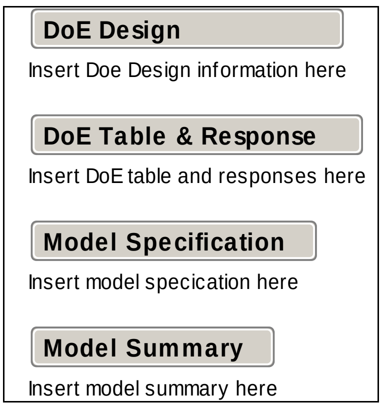



Figure 4.5.7 Template JMP journal file

The journal feature in JMP allows information to be copied to a log file when the user instructs to do so, and the template provides a standard index to remind the user to fill in the content. With the previous imaginary sample problem, all the modeling information can be recorded in a journal file as illustrated in Figure 4.5.8 to Figure 4.5.12. The journal file includes the several sections.

The DoE design section, as displayed in Figure 4.5.8, tracks the output variables (responses) with names, limits, and the goals of optimization, input variables (factors) with ranges and characteristics, and the type of the DoE with related parameters for the generation of the design of experiment.

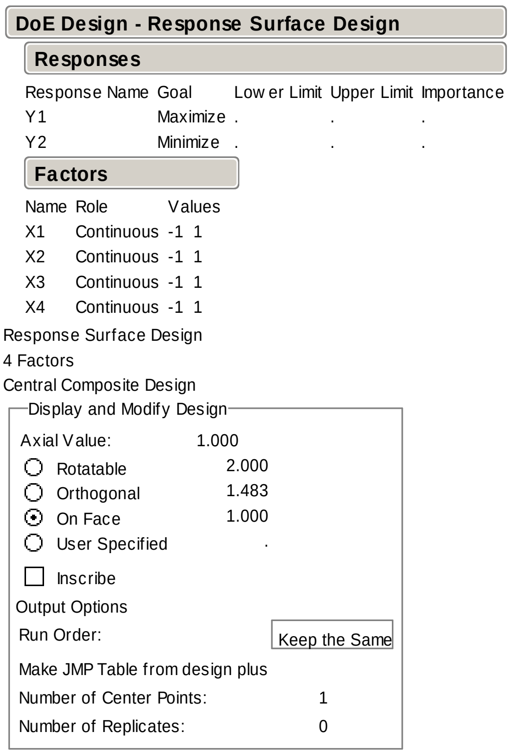



Figure 4.5.8 Sample DoE Design

The DoE Table & Responses section logs the complete DoE cases and the responses related for each of the cases. Figure 4.5.9 shows a typical DoE setting. Each row in the table represents a case run with values of the input variables and the responses variables.

**DoE Table & Response**

| Pattern | X1 | X2 | X3 | X4 | Y1 | Y2 |
|---|---|---|---|---|---|---|
| −−−− | -1 | -1 | -1 | -1 | 1.67880838 | 1.62064963 |
| −−−+ | -1 | -1 | -1 | 1 | 0.89482357 | 0.79164673 |
| −−+− | -1 | -1 | 1 | -1 | 1.1967852 | 1.39808699 |
| −−++ | -1 | -1 | 1 | 1 | 0.59930595 | 0.54689924 |
| −+−− | -1 | 1 | -1 | -1 | 2.86477482 | 1.01685436 |
| −+−+ | -1 | 1 | -1 | 1 | 1.78116903 | 0.88557873 |
| −++− | -1 | 1 | 1 | -1 | 1.58477917 | 1.82235579 |
| −+++ | -1 | 1 | 1 | 1 | 1.773269 | 0.31582867 |
| +−−− | 1 | -1 | -1 | -1 | 0.40789718 | 0.54894018 |
| +−−+ | 1 | -1 | -1 | 1 | 0.86527043 | 0.85978314 |
| +−+− | 1 | -1 | 1 | -1 | 0.27400989 | 0.01845087 |
| +−++ | 1 | -1 | 1 | 1 | 0.00085612 | 0.90033108 |
| ++−− | 1 | 1 | -1 | -1 | 1.31294415 | 0.77456831 |
| ++−+ | 1 | 1 | -1 | 1 | 1.20137016 | 1.15824672 |
| +++− | 1 | 1 | 1 | -1 | 2.00071286 | 0.47718237 |
| ++++ | 1 | 1 | 1 | 1 | 1.75745762 | 0.03379191 |
| a000 | -1 | 0 | 0 | 0 | 1.51219399 | 0.49366198 |
| A000 | 1 | 0 | 0 | 0 | 1.00372838 | 0.65282429 |
| 0a00 | 0 | -1 | 0 | 0 | 0.0117263 | 0.1958893 |
| 0A00 | 0 | 1 | 0 | 0 | 0.5622159 | 0.3840429 |
| 00a0 | 0 | 0 | -1 | 0 | 0.74555007 | 0.06857882 |
| 00A0 | 0 | 0 | 1 | 0 | 0.08690954 | 0.00335915 |
| 000a | 0 | 0 | 0 | -1 | 1.40348191 | 0.53119154 |
| 000A | 0 | 0 | 0 | 1 | 0.00997321 | 0.99489828 |
| 0000 | 0 | 0 | 0 | 0 | 0.23151874 | 1.43449742 |

Figure 4.5.9 Sample DoE Table and Responses

The model specification section in Figure 4.5.10 preserves the characteristics of the model and the model construction elements that are created with the combination of input variables.

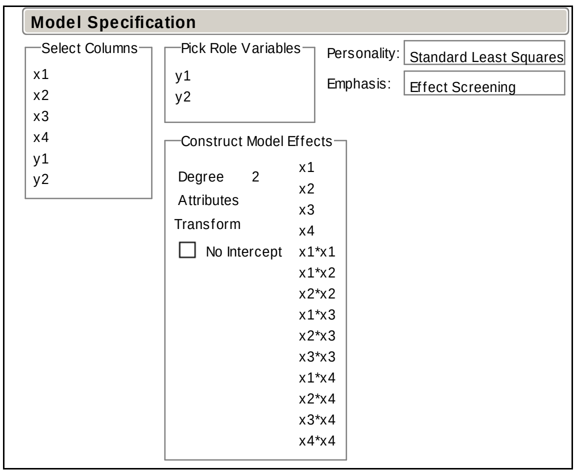
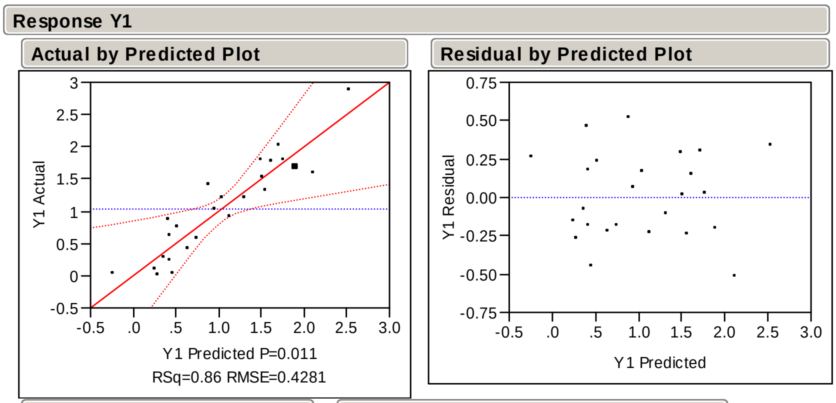



Figure 4.5.10 Sample Model Specification

Illustrated in Figure 4.5.11, the model summary section keeps the model itself. It includes the following subsections. Parameter estimates provides the coefficient of each model term. Fitting summary depicts the accuracy of the model, such as $R^2$. Row diagnostics visualizes the relationships between observed values and predicted values, and magnitude of the difference, such as Actual by Predicted Plot and the Residual by Predicted Plot.




**Summary of Fit**

| | |
|---|---|
| RSquare | 0.862704 |
| RSquare Adj | 0.67049 |
| Root Mean Square Error | 0.428058 |
| Mean of Response | 1.030461 |
| Observations (or Sum Wgts) | 25 |

**Parameter Estimates**

| Term | Estimate | Std Error | t Ratio | Prob>\|t\| |
|---|---|---|---|---|
| Intercept | 0.4243849 | 0.18483 | 2.30 | 0.0446* |
| X1 | -0.281203 | 0.100894 | -2.79 | 0.0192* |
| X2 | 0.4949561 | 0.100894 | 4.91 | 0.0006* |
| X3 | -0.137696 | 0.100894 | -1.36 | 0.2022 |
| X4 | -0.213372 | 0.100894 | -2.11 | 0.0606 |
| X1*X2 | 0.0681364 | 0.107014 | 0.64 | 0.5386 |
| X1*X3 | 0.1444369 | 0.107014 | 1.35 | 0.2069 |
| X2*X3 | 0.1082377 | 0.107014 | 1.01 | 0.3357 |
| X1*X4 | 0.1316231 | 0.107014 | 1.23 | 0.2469 |
| X2*X4 | -0.003294 | 0.107014 | -0.03 | 0.9761 |
| X3*X4 | 0.0372746 | 0.107014 | 0.35 | 0.7348 |
| X1*X1 | 0.801432 | 0.26823 | 2.99 | 0.0136* |
| X2*X2 | -0.169558 | 0.26823 | -0.63 | 0.5415 |
| X3*X3 | -0.040299 | 0.26823 | -0.15 | 0.8836 |
| X4*X4 | 0.2501983 | 0.26823 | 0.93 | 0.3729 |

Figure 4.5.11 Sample of Model Summary for a Response

As displayed in Figure 4.5.12, a prediction profiler allows users to visualize and interactively investigate the design space, such as predicting the values of the output variables for any combination of the input variables, visualizing slices of the resulting design space, and manually exploring and searching the design space for good designs.

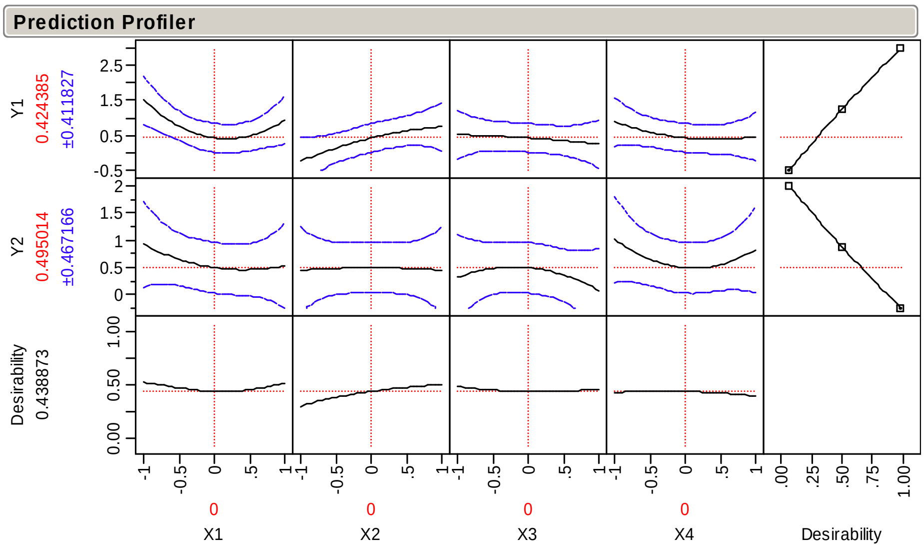



Figure 4.5.12 Sample of Recorded Prediction Profiler

On the other hand, we also store some important information about the model in Excel for portability, such as DoE setting and the model itself (Figure 4.5.13 and 4.5.14)

**DoE**

| | |
|---|---|
| Doe Design | Central Composite Design |
| Number of Factors | 4 |
| Number of Runs | 25 |
| Block Size | |
| Center Points | 0 |
| Axial Value | 1 |

| Case | X1 | X2 | X3 | X4 |
|---|---|---|---|---|
| 1 | -1 | -1 | -1 | -1 |
| 2 | -1 | -1 | -1 | 1 |
| 3 | -1 | -1 | 1 | -1 |
| 4 | -1 | -1 | 1 | 1 |
| … | | | | |

Figure 4.5.13 Sample DoE Setting

**RSE**

| | |
|---|---|
| R square | 0.86 |

| Term | Estimate | Std Error | t Ratio | Prob>\|t\| |
|---|---|---|---|---|
| Intercept | 0.4243849 | 0.1848301 | 2.2960809 | 0.0445504 |
| X1 | -0.2812035 | 0.1008942 | -2.7871128 | 0.0192147 |
| … | | | | |

Figure 4.5.14 Sample Response Surface Model

As a summary, this sub-chapter describes the approach of maintaining knowledge integrity for surrogate models. It establishes standard information templates with defined data structure, and provides a process to guide the user retaining the valuable data and model information throughout the modeling process. With this approach, the model creator or other designers can track back the model generation process, make updates to the model when necessary, and create a new model. It significantly improves the reusability of the surrogate models.

## 4.6 Knowledge Presentation for Aircraft Life cycle Design Decision Support

### 4.6.1 Knowledge Maps of the Aircraft Life cycle

As discussed in the motivation section, the phases in an aircraft life cycle are usually separated, and aimed for different objectives. Focusing on one or two dimensions of the aircraft life cycle might be efficient to achieve quick results, but the results are local optimums due to the lack of information from other dimensions, or not optimal at all when other considerations are taken into account. Therefore, a knowledge map system is proposed to support decision making in the whole aircraft life cycle, which is a modular structure, and can include both engineering and business aspects, Figure 4.6.1.

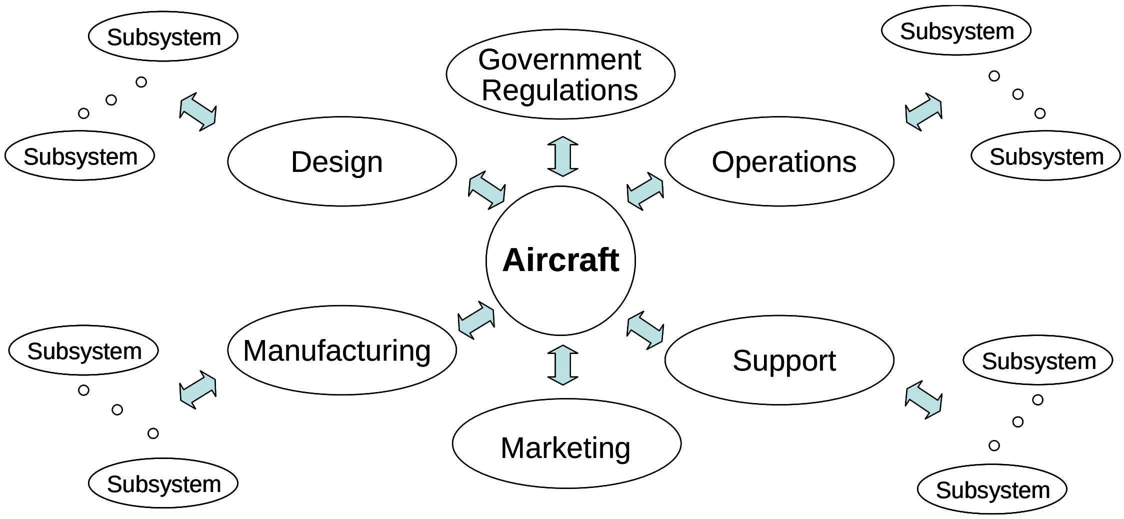



Figure 4.6.1 Knowledge Presentation for Aircraft Life cycle Decision Support

Within the framework, all aircraft related activities can be included so that designers will have a complete view on all the aspects of the aircraft life cycle. The knowledge map can be used as a central access point for all the aircraft life cycle related knowledge. From this map, a user can select to go deeper into a branch to view more detailed knowledge when making the design decision. For example, if he is interested in the design technology alternatives in the conceptual design phase, he can follow the link to a system or subsystem to display the morphological matrix associated with this aircraft type and view all the listed technologies for each of the functional area. If he is not sure about a technology concept, he can click on the technology, and get more detailed information and explanation.

In addition, the information is presented in a hierarchical style, where details of information are encapsulated in different levels, so that decision makers can view the information at the level of details tailored to their specific interest. For decision making at a higher level, the information can be in a concise and easy-to-understand format, so that they can just take the results and do not have to spend too much time to figure out what is going on. For detailed engineering design, the information can be in a detailed format with supporting data, so that they will know how those results are created and be able to verify the results by themselves.

### 4.6.2 Design Space Exploration with Knowledge-enabled Multimedia Morphological Matrix

In the aircraft life cycle, the concept generation phase is an important stage of the aircraft design process. This phase dictates the level of innovation and also predetermines the aircraft cost to a significant extent. With the importance of the concept generation phase, researches have been carried out on creating generic methodologies for improving the innovative abilities of the designer [Pahl 1996, Jansson 1990]. By dividing the design task into smaller tasks, these methodologies try to narrow the cognitive effort and focus on the innovative thinking; and then, to create several solutions to address each function. Once the solutions are generated, an overall solution is synthesized by identifying individual solutions for each function that is compatible. This is the core of the morphological matrix.

Generating concepts from a morphological matrix started sixty years ago [Zwicky 1948], and it is still widely used today as an important step in the engineering design process [Hubka 1988, Pahl 1996, Ullman 1997]. The morphological matrix is a methodology for organizing alternative solutions for each function of a system and combining them to generate a great number of solution variants each of which can potentially satisfy the system level design need.

A morphological matrix looks like a table, and consists of a grid of columns and rows. The first column lists the relevant functions, which is a functional breakdown of the whole system or subsystem, and the row adjacent to each function lists the possible solutions that will achieve the function. In developing the matrix, the designers use text to represent the solutions. Once the matrix is created, the designer can pick one solution for each function row, and combine the individual solutions into an effective conceptual design. If the designer repeats the process, a pool of conceptual designs can be generated for further investigation.

The morphological matrix methodology is an excellent way to record the solutions for the relevant functions and aid in the cognitive process of generating the system-level design solutions. However, the traditional morphological matrix has a significant limitation, which is the user of the matrix has to be familiar with the individual solutions to be able to select them wisely. The text in each of the cell does not contain enough information. Ideally, if we can assume the user is an expert in all the aspects of the design domain, and knows every technical detail of each solution, it might not be a problem. In reality, this is not the case. Although the user may be an expert in one particular area, he/she may still lack specific knowledge related with certain solutions in the matrix. Without knowing the advantages and disadvantages of each solution and the potential effect the solution applying on other related functions, it will be impractical to select the combination of the functional solutions effectively. For example, if one does not know solution A1 for function A is not compatible with solution B2 for function B, and selected the combination of A1 and B2 in the design, then the results will be an invalid design. Therefore, how to make the morphological matrix more informative becomes a critical issue for this methodology.

To infuse knowledge into the morphological matrix and make it more practical to the users, we propose the knowledge-based morphological matrix.

With the current information technology, we can easily manage and present more information to decision maker. The technologies we used here combined the feature of the morphological matrix with multimedia and hyperlinks.

With multimedia, knowledge can be presented to all human cognitive channels, such as photos, audio, video, and even interactive features. For example, if one does not know what the turbine engine is, a movie of turbine engine introduction or an animation of turbine engine at work will definitely make him/her quickly grasp the idea.

With hyperlink, knowledge can be presented in an organized and easy-to-access structure. The resources are available to the user, and can be accessed with a single mouse click.

A sample knowledge-based morphological matrix for automotive design is illustrated in figure 4.6.2.

| | Functional Decompositions | Alternative 1 | Alternative 2 | Alternative 3 | Alternative 4 | Alternative 5 |
|---|---|---|---|---|---|---|
| Engines | Engine Type | Gasoline | Steam | Electric | Hybrid | Fuel Cell |
| | Cylinder Number | 2 | 4 | 6 | V8 | V16 |
| Body | Skin | Steel | Fiberglass | Aluminum | Carbon Fiber | |
| | Skeleton | Chassis | Unitized Body | Rolling Chassis | | |
| Drive Systems | Suspension | Spring & Shock Absorber | Attach wheels to skeleton | Independent Suspension System | Active Suspension System | |
| | Transmission | Belts / Chains | Drive Shaft | Manual | Auto | Continuous |
| | Drive Wheel | Front | Rear | | | |
| | Steering Wheel | Tilter | Gear & Rack | Power Steering | Electric Steering | |
| | Starter | Crank | Electronic | | | |
| | Battery Power | 6V | 12V | 42V | | |
| Wheels | Tire Structure | Solid Rubber | Inner Tube | Tubeless | | |
| | Tire Ply | Bias | Radial | | | |
| | Thread | No Thread | Thread | | | |
| | Tire Material | Natural rubber | Synthetic | Natural & Synthetic | | |
| | Brakes | Pure Friction | Disk/Caliper | Drum | ABS | Disk+Drum+ABS |
| Accessory | Inside Temperature Control | Exhaust Fume | Semi-portable Hot Water Heater | Nash System | A/C | Nash+A/C |
| | Seat | Fixed Wooden Bench | Adjustable Front Seat | | | |
| | Lock | Door Lock | Ignition Lock | | | |
| Safety | Airbags | Driver Side | Passenger Side & Side Impact | Head Curtain | Smarter Airbag | |

Figure 4.6.2 Sample knowledge-based morphological matrix

# CHAPTER V
# IMPLEMENTATION

## 5.1 Feature-Based Hierarchical Aircraft Maintenance Knowledge Discovery

An aircraft maintenance database in an airline updates daily to track each of the services for each of the aircraft. Therefore, it contains significant amount of data and keep growing. The goal is to discover possible knowledge to help improve the aircraft maintenance processes, some potential benefits gain from the framework are:

- Reduce flight schedule disruptions
- Reduce the number of unscheduled maintenance events
- Make aircraft more robust and easier to maintain by giving suggestions to the aircraft manufacturers to influence future design requirements
- Reduce maintenance service durations
- Potential predict unscheduled maintenance events to improve recovery capability
- Lower maintenance cost.

In the given database, there are about half a million entries of aircraft daily maintenance data, which were recorded during 10 months of operation for a major domestic airline. Although it contains only data from a certain period, it may be treated as a snapshot of the airline's day-to-day maintenance, and reflects typical service operation.

### 5.1.1 Knowledge Discovery Goal

In this example, after brain storming, the goal is defined to discover knowledge, in the given aircraft maintenance database, to provide suggestions and recommendations on improving airline unscheduled maintenance operation.

### 5.1.2 Data Preparation

Data preparation is to trim and clean data according to the overall goal.

The original database contains two overall categories of maintenance operations, i.e. scheduled maintenance and unscheduled maintenance. Scheduled maintenances are planed in advance to keep aircraft in good operating condition, which a proactive approach to maintenance and will not disrupt the airline's normal flight schedule. On the other hand, unscheduled maintenances are unpredictable in advance, and are consequence of certain abnormities in regular operation. Although some of them can be fixed quickly without any disruption, most of the services will have to request the normal flight schedule to be altered (delayed or even canceled) since they are found right before the scheduled flight. Due to the nature of unscheduled maintenance actions, when they occur, they cause a significant negative impact on the airline's regular operation, and thus, it is expected that they are minimized or eliminated when possible. Therefore, in this study emphasis is given on unscheduled maintenance, while scheduled maintenance is not considered. We use the following query to filter out the scheduled maintenance records.

```
Select * into local_data_unscheduled
From   local_data
Where  time_control = false
```

After this simple filtering operation, the number of records dropped from the original 0.5 million to 0.26 million. The size of the dataset we are facing is reduced by one half, which makes the investigation easier. Moreover, it demonstrates that about half of the maintenance records are disruptive unscheduled services, which confirms the importance of our knowledge discovery goal.

### 5.1.3 Feature Exploration

5.1.3.1 Potential Feature Generation

This step does not depend on the data cleaning process; therefore, it doesn't have to be done after the data preparation.

***A. Low-Level Feature***

User-based features: as the nature of this exploration, we assume the user has some basic knowledge on aircraft maintenance, and the user has identified some features as follows:

*Maintenance duration*, which is the duration for a particular service to be done.

*Frequency*, which is how many times a particular type of service happens in a given time period.

Data structure based features: some features are identified by selecting the table column names existing in the original database. For example,

*LOG_DT*, which is the date and time when the service was performed.

*SHIP_NBR*, which is a number that uniquely identifies an aircraft

*STN_AT_OMI_COM*, which is the abbreviated name of the service station.

*CAR_RMKS*, which is the remarks associated to a particular service.

*BB_TASK_ID*, which is a unique identifier of a certain type of service task. One task may be carried out many times

*FLEET*, which is a unique identifier of a certain type of aircraft

*MAN_HOURS*, which is how many hours it takes a particular service to be done.

As one can see there are some overlaps between the above two categories, which is reasonable since the database records some basic operation aspects. For instance, maintenance duration is represented as manhours in the database. Combining them by removing the duplicates, we have the following pool of low-level features:

| | | | |
|---|---|---|---|
| *Frequency* | *man_hours* | *log_dt* | *ship_nbr* |
| *stn_at_omi_com* | *car_rmks* | *bb_task_id* | *fleet* |

***B. Mid-level Feature***

By mimicking the real world situation, it is assumed that some domain experts are involved to identify the mid-level features (Expert-based features).

From *domain knowledge*, some features are obtained about the general maintenance process as follows

*MTBF* (Mean Time Between Failures), which represents the average time passed between to consecutive failures. It can be task specific, which only counts the time based on a specific task, fleet specific, or ship specific.

MTBF is a basic measure of reliability for repairable items. It can be described as the number of hours that pass before a component, assembly, or system fails. It is a commonly used variable in reliability and maintainability analyses.

MTBF can be calculated as the inverse of the failure rate for constant failure rate systems. For example: If a component has a failure rate of 2 failures per million hours, the MTBF would be the inverse of that failure rate.

MTBF = (1,000,000 hours) / (2 failures) = 500,000 hours

The focus is on measuring how an aircraft, or a fleet of aircraft behave on certain tasks. Therefore, we have the following mid-level features based on the *problem background*:

*Task*, which identifies a certain type of service jobs. One task may be carried out many times. To avoid confusion, we use the name *task* for a type of task, and the name *job* for a instance of a task.

*Fleet,* which identifies a certain type of aircraft

*Aircraft,* which identifies a particular aircraft, such as a tail number.

***C. High-Level Feature***

Features on this level are not obvious, and may require lots of computation to obtain them. However, they may be more critical than other levels in knowledge discovery. High-level features are based on the low-level and mid-level features, and may be in the forms of combinations or derivatives of them. Some approaches on creating potential high-level features are:

<u>Arithmetic-based feature exploration</u>: Some basic arithmetical operators, such as summation, average and count, are used on certain existing features to aggregate information to a higher level. The new features are as the first column in Table 5.1.1

Taking into the different sizes of fleet, we normalized some features by dividing number of ships for each fleet. After we reword some names of features, we have the modified features in second column in Table 5.1.1.

Table 5.1.1 Arithmetic-based Features

| Original Features | Modified Features |
|---|---|
| Sum of man_hours by bb_task_id | TotalTaskManHours/Ship |
| Sum of man_hours by fleet | TotalFleetManHours/Ship |
| Sum of man_hours by ship_nbr | TotalShipManHours |
| Sum of man_hours by stn_at_omi_com | TotalStationManHours |
| Count of bb_task_id by fleet | FleetTaskCount |
| Count of ctrl_nbr by bb_task_id | TaskFreq/Ship |
| Count of ctrl_nbr by fleet | FleetServiceFreq/Ship |
| Count of ctrl_nbr by ship_nbr | ShipServiceFreq |
| Average of man_hours by bb_task_id | AvgTaskHours |
| Average of man_hours by fleet | AvgFleetHours |
| Average of man_hours by ship_nbr | AvgShipHours |

Statistics-based feature exploration: Taking a further step, statistical formulas can be used to help generating high level features out of existing features.

For instance, one can use histograms, which calculate individual and cumulative *frequencies* on *AvgTaskHours* for each of the fleet. Since we have discrete data set, we divide *AvgTaskHours* into buckets, and sum the data in each of the buckets. Thus, we obtain a new feature, *Freq_AvgTaskHoursInFleet*, which will show number of occurrences in each of the *AvgTaskHours* bucket in a fleet. Cumulative frequency is calculated by summing all the occurrences where the *AvgTaskHours* is less than the current bucket value. For example, we can divide the *AvgTaskHours* into 0-1, 1-2, 2-3, 3-4, 4-5, 5-10, 10+ seven buckets, and a task with *AvgTaskHours*=1.5 will fall in 1-2 bucket. In fleet B777, there are 163 tasks whose average service man hours are between 0 to 1 hour, and it takes about 59% of total number of tasks.

Moreover, one can use some other features, instead of *frequency*, as merits to show other aspects of information in *AvgTaskHours*. If we use *TotalTaskManHours/Ship* on *AvgTaskHours*, some new features are listed as follows:

*TotalTaskManHours/Ship_AvgTaskHourInFleet*

*CummulativeTotalTaskManHours/Ship_AvgTaskHourInFleet.*

With similar pattern, we can have

*TaskFreq/Ship_AvgTaskHourInFleet*

*CummulativeTaskFreq/Ship_AvgTaskHourInFleet*

Taking it one step further, we can replace *AvgTaskHours* with other features, for example, *TaskFreq/Ship.* Thus we'll have new features as follows:

*Freq_TaskFreq/ShipInFleet*

*CummulativeFreq_TaskFreq/ShipInFleet.*

Replacing frequency with TotalTaskManHours/Ship, we have

*TotalTaskManHours/Ship_TaskFreq/ShipInFleet*

*CummulativeTotalTaskManHours/Ship_TaskFreq/ShipInFleet*

As one can see, the features are getting more complex, and they become more abstract, and closer to the knowledge we are seeking.

<u>Function-based feature exploration</u>: Certain distribution functions, such as normal, Weibull, and exponential, might be able to create features with their parameters, if they are assumed to be the distribution of certain data. In addition, normalization could be used to compare apples to apples. The detailed implementation is to be explored.

<u>Logics-based exploration</u>: Since there are too many tasks listed in the database, we may set our focus on important tasks, which could be in one of the following two categories:

Some tasks take long time and a great effort to finish, although they do not occur often in the daily services. This type of tasks could post a big impact on regular flight schedules. We use the following logic expression as a feature to facilitate our investigation.

$$AvgTaskHours > \alpha$$

Tasks happen very often. Although individual task may not take too long, it's become a bigger concern if it occurs in a high frequency. It naturally leads us to a new feature.

$$TaskFreq > \beta$$

However, this feature could mislead us since the fleets have different number of ships. Therefore, we redefine the feature to a new formula which divides the frequency by the number of ships in each fleet, and removes the fleet size issue.

$$TaskFreq/Ship > \beta$$

Although we have the above features, one question may arise: What are the values of $\alpha$ and $\beta$? This is a topic to be discussed in the next section.

5.1.3.2 Feature Identification

As for the variables in the features, such as $\alpha$ and $\beta$ mentioned in feature generation, we need an algorithm to decide the best values for them. If we are using the features to seek abnormality in the data set, we can look at classification and find out which classes of data represent significant difference to other classes. One of the performance measuring criteria is the significance of classification effects on the given data set. We could use the standard deviation, which is a measure of how widely values are dispersed from the average value (the mean). However, it may be moderate if there are only a small portion of outliners. Since we are seeking outliers, we use Range/Mean to evaluate the performance of a feature, where Range is Max-Min.

The pseudo code for the algorithm is as follows.

```
For α from 0 to 5 increment by 1
  For β from 0 to 5 increment by 1

    Select
      Sum(TaskFreqPerShip) AS TotalTaskFreqPerShip,
      Sum(TotalTaskManHoursPerShip) AS TotalTaskHrPerShip
    From taskStatsFromLocalData
    Where
   AvgTaskHours > α
   and TaskFreq/Ship > β
    Group by fleet

   maxFreq=max(TotalTaskFreqPerShip)
   minFreq=min(TotalTaskFreqPerShip)
   avgFreq=average(TotalTaskFreqPerShip)
   evaFreq=(maxFreq-minFreq)/avgFreq

   maxHr=max(TotalTaskHrPerShip)
   minHr=min(TotalTaskHrPerShip)
   avgHr=average(TotalTaskHrPerShip)
   evaHr=(maxHr-minHr)/avgHr

   store(α, β, evaFreq, evaHr) into array-A

  End for
End for

Select max(evaFreq) from array-A
and set α1=α, β1=β
Select max(evaHr) from array-A
and set α2=α, β2=β
```

From the above algorithm, there are two sets of $\alpha$ and $\beta$, where $\alpha 1$ and $\beta 1$ correspond to the data with the biggest gaps between fleets on *TaskFreq/Ship*, while $\alpha 2$ and $\beta 2$ correspond to the data with the biggest gaps between fleets on *TotalTaskManHours/Ship*.

### 5.1.4 Feature-Based Knowledge Mining and Result Visualization

Once we have a pool of key features, from low-level to mid-level, to high level, we can start to investigate the data set with those features.

Some sample mining results are demonstrated below. We use a hierarchical approach to start from top level comparison of the service levels between fleets, and then focus on one fleet and compare tasks categories within the fleet, and then investigate one specific task, see Figure 5.1.1.

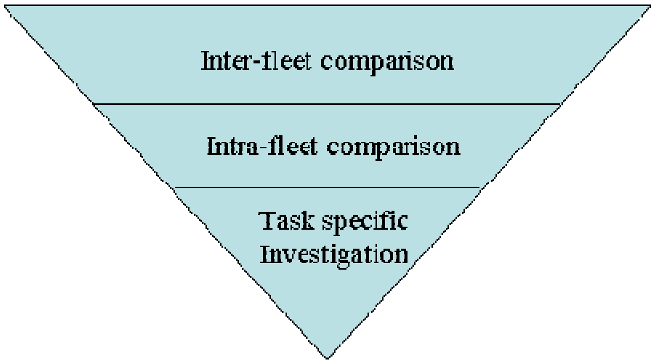



Figure 5.1.1 Hierarchical Knowledge Discovery

5.1.4.1 Inter-Fleet service comparison

Using the feature *Fleet* as a base, and *FleetServiceFreq/Ship*, *TotalFleetManHours/Ship*, and *AvgFleetHours* as a measurement of merit, and use AvgTaskHours > 2 and TaskFreq/Ship > 1 as filter to set our focus on relatively longer and oftener tasks, we can compare the service frequency, total man hours and average service duration between fleets, see Table 5.1.2.

Table 5.1.2 Inter-fleet Service Comparison

| FLEET | FleetServiceFreq/Ship | TotalFleetManHours/Ship | AvgFleetHours |
|---|---|---|---|
| B732 | 80.17 | 287.20 | 3.58 |
| B733 | 70.04 | 341.40 | 4.87 |
| B738 | 17.83 | 81.38 | 4.56 |
| B757 | 27.45 | 92.12 | 3.36 |
| B764 | 49.10 | 156.82 | 3.19 |
| B76D | 86.71 | 281.70 | 3.25 |
| B76L | 32.00 | 105.09 | 3.28 |
| B777 | 225.88 | 557.14 | 2.47 |
| MD88 | 54.64 | 172.53 | 3.16 |
| MD90 | 86.88 | 384.97 | 4.43 |

From the result, we have the following observations, and some knowledge is extracted from the observation, and some suggestions are provided to aid business decision making, see Table 5.1.3. Further interpretation may request domain expert's involvement.

Table 5.1.3 Observations and Suggestions

| Observations | Knowledge and Suggestions |
|---|---|
| B777 has much higher unit service frequency around 226 times, which results much higher total service hours at around 557 hours. | B777 is relatively new, and has more features and equipments. An advanced fleet does not mean a reliable fleet. Reasons of high service frequency are worthy of further investigation. |
| B738, B757, and B76L have much less service occurrence. | Those types of aircrafts were tested over time, and could be considered stable. |
| B777 has lowest average service duration below 3 hours. | Although the B777 fleet has more problems, they are relatively easy to fix. This might be a consequence of better design practices upfront. |
| B733, B738 and MD90 takes longer, over 4 hours, to repair | Some older models of aircraft might be harder to fix |
| B757 and B76L are relatively healthy and easy to repair | Although they were in service for long time, it might be worth to keep them due to low maintenance |

To better represent the results and make easier comparison, we can visualize the results by combining them into figures, see Figure 5.1.2. In the left figure, the number of services, which are longer than 2 hours and happen over 1 time per ship on average, is displayed with light blue bars; and total service man hours spent on those tasks for each fleet are displayed with dark purple bars. In the right figure, average man hours per specified task are displayed in light blue for each fleet.

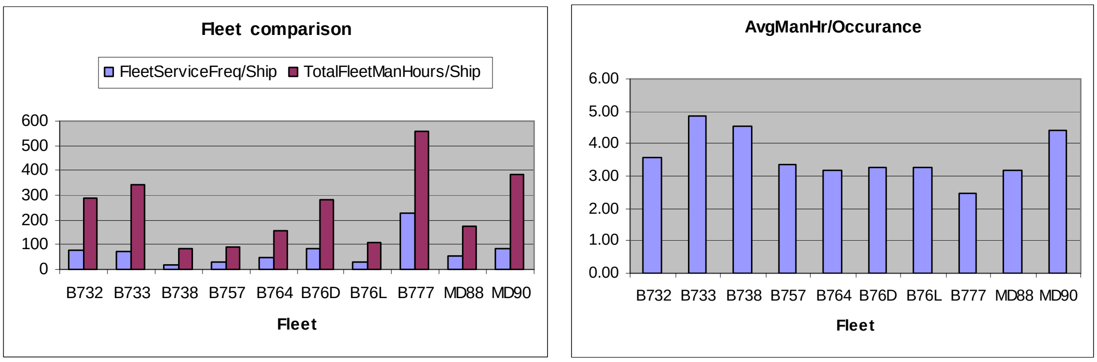



Figure 5.1.2 Fleet Comparison

5.1.4.2 Intra-Fleet overall task characteristics exploration

In the above section, we compared fleet service characteristic at top level. Take it one-step further now to review the overall task characteristics in a fleet. It is found that B777 is a problem fleet, so the focus is on the B777 and see if some in-depth knowledge can be found.

Use the features we identified in chapter 3, we first take a look at *Frequencies* and their cumulative effect in each *AvgTaskHours* bucket.

Table 5.1.4 Intra-fleet Overall Task Characteristics in AvgTaskHours Bucket

| Avg Hrs | # of Tasks | Cumulative | TotalHrs/Ship | Cumulative | Occurs/Ship | Cumulative |
|---|---|---|---|---|---|---|
| 1 | 109 | 39.49% | 222.05 | 16.18% | 272.13 | 30.91% |
| 2 | 112 | 80.07% | 470.08 | 50.43% | 363.88 | 72.24% |
| 3 | 24 | 88.77% | 433.60 | 82.03% | 210.00 | 96.10% |
| 4 | 10 | 92.39% | 43.26 | 85.18% | 12.38 | 97.50% |
| 5 | 4 | 93.84% | 8.85 | 85.82% | 2.00 | 97.73% |
| 10 | 10 | 97.46% | 95.98 | 92.82% | 13.38 | 99.25% |
| 10+ | 7 | 100.00% | 98.59 | 100.00% | 6.63 | 100.00% |
| Total | 276 | | 1372.40 | | 880.38 | |

From Table 5.1.4, we find that,

- In the total 276 tasks of B777, majority of tasks (80%) are relatively short tasks, whose average man hours are shorter than 2 hours;
- 82% of the service hours are spent on the tasks whose average man-hours are between 1 hour and 3 hours.
- Most individual jobs (96.10%) are shorter than 3 hours.

Numbers are visualized in Figure 5.1.3 to facilitate direct comparison.

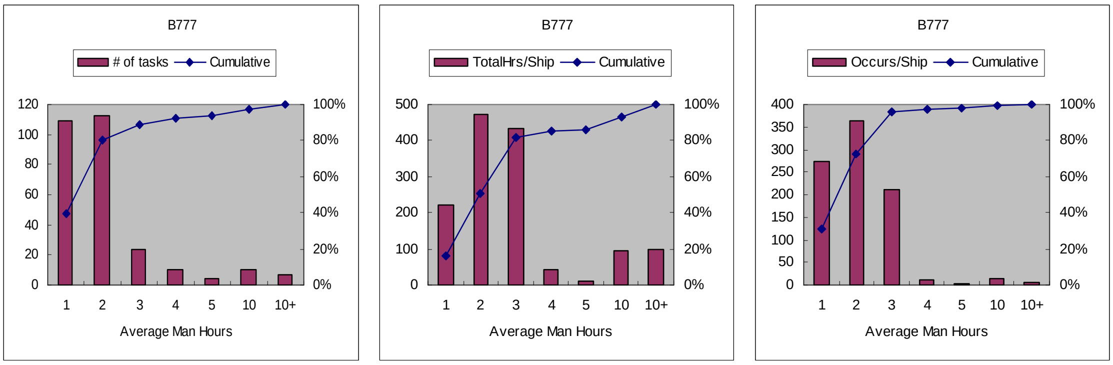



Figure 5.1.3 Visualization of Information in AvgTaskHours Bucket

Besides the *AvgTaskHours* bucket, we can also use the other bucket features. For example, *TaskFreq/Ship* can be used to reveal information of another aspect.

Table 5.1.5 Task Frequency Investigation

| TaskFreq/Ship | # of tasks | Cumulative | TotalHrs/Ship | Cumulative |
|---|---|---|---|---|
| 1 | 153 | 55.43% | 178.81 | 13.03% |
| 2 | 38 | 69.20% | 93.25 | 19.82% |
| 3 | 23 | 77.54% | 115.35 | 28.23% |
| 5 | 25 | 86.59% | 114.83 | 36.60% |
| 10 | 23 | 94.93% | 229.48 | 53.32% |
| 50 | 12 | 99.28% | 208.50 | 68.51% |
| 50+ | 2 | 100.00% | 432.19 | 100.00% |

From Table 5.1.5, we find that,

- Most of the tasks do not occur often. Among them, over one half of the tasks occur less than one time in the 10 month period; and about 87% of tasks occur less than 5 times.
- A large portion of man-hours, about 47% (=1-53%), is spent on the frequent tasks, whose frequencies are over 5 times per ship, although there are only 5% (=1-95%) tasks in this category.

These results are visualized in Figure 5.1.4 and they are presented side by side to facilitate a direct comparison.

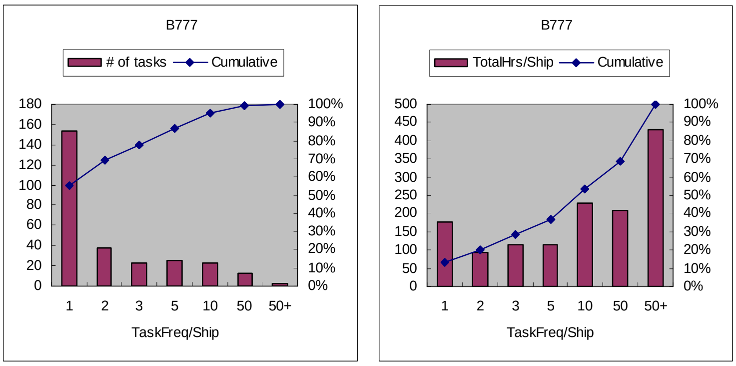



Figure 5.1.4 Comparison of Task Properties for B777

5.1.4.3 Inter-fleet comparison

Besides going deep into specific task as in section II, we can do another type of inter-fleet comparison following the direction of section II. We picked B737, B757, B767, and B777 as samples to compare, and use charts to visualize the results, see Figure 5.1.5.

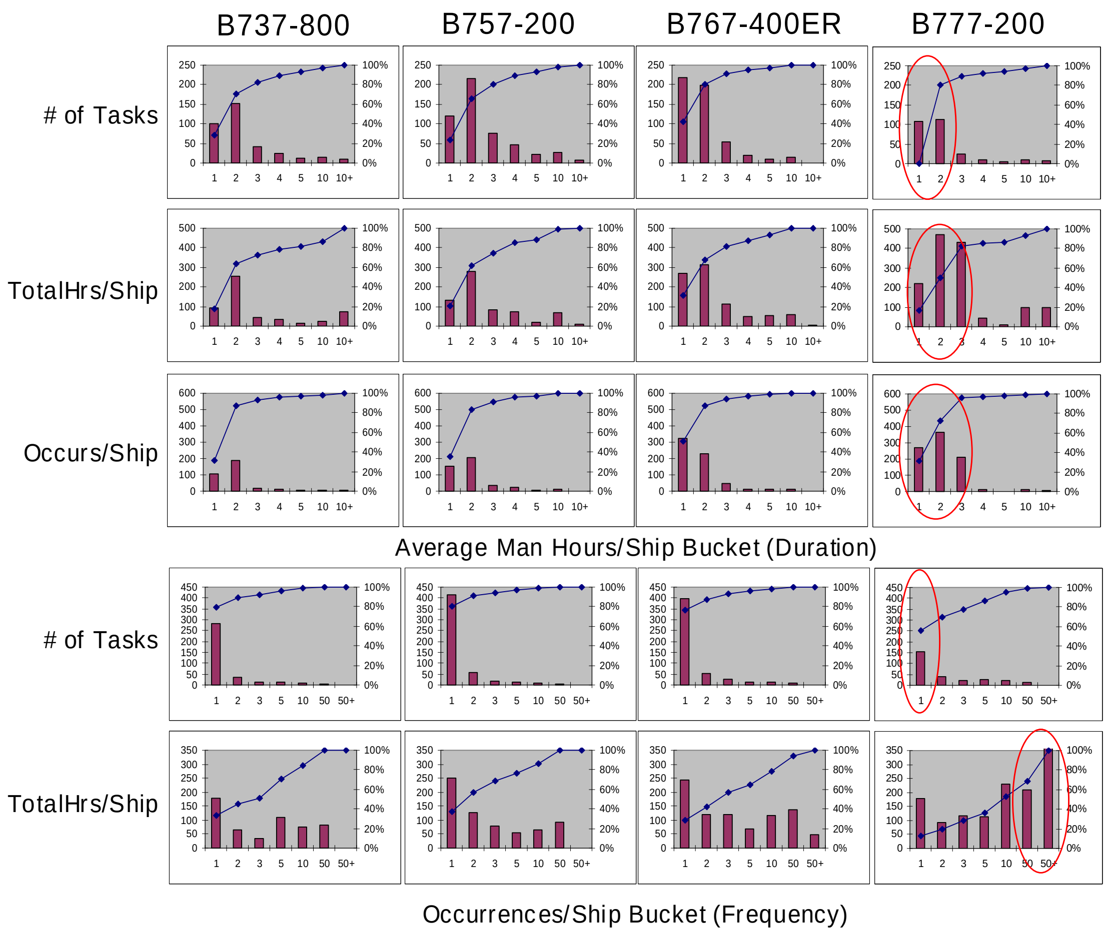



Figure 5.1.5. Characteristics Comparison between Fleets

In the above matrix of figures, the first three rows have the same abscissas as AverageManHours/Ship bucket; and the next two rows have the same abscissas as Occurences/Ship bucket. Each column represents a fleet. With results on same scales across fleets, one can easily exam certain feature sets between fleets by comparing them column-wise. For example, B777 is different to others in the following aspects:

Row one: it has less number of tasks, which is represented by less covered areas under red; and majority (80%) of tasks are short tasks, which is shown by the data points connected by lines.

Row two: It has more service hours per ship, and longer total service hours in shorter tasks. The observations between row one and row two seems contradictory, and the results on row three explain it.

Row three: Tasks in B777 are more often to occur than other fleets, and the shorter tasks take more than 95% of the occurrence.

Row four: B777 has less number of infrequent tasks (70%), while other fleets have around 90% tasks on low frequency tasks, which have occurrence less than twice per ship.

Row five: B777 takes more maintenance hours on high frequency tasks. Around 65% of time was spent on common tasks, whose occurrence is more than 50 times per ship. On the other hand, other fleets have less than 5% of the time spent on those types of tasks.

As an alternative to the above approach, a more graphics-oriented hierarchical system knowledge exploration methodology is investigated and illustrated in chapter 5.2

## 5.2 System Analysis with Bottom-up and Top-down Approaches

In this sub-chapter, we have two examples to illustrate the feasibility of the system analysis approaches. First, we investigate the aircraft maintenance system, and then we move our focus onto the analysis of airline operations.

### 5.2.1 System Analysis on Aircraft Maintenance System

An aircraft itself is a complex system with hundreds of subsystems and tens of thousands of parts. It requires significant effort for aircraft designer to design high quality aircraft and for airlines to maintain the good shape of hundreds of aircraft in different fleets. In this section, we will examine the maintenance log from a major domestic airline. The period of consideration is from February 22nd to December 15th 2003. During this ten month period half a million records of data were collected and analyzed to identify potential aircraft design problems attributable to maintenance issues. The intent was to provide feedback to the aircraft designers so that the aircraft design can be improved from a reliability and maintainability perspective.

5.2.1.1 Preliminary Analysis

With the huge amount of data available, one could be easily buried by the overwhelming information. To better understand the data, we first need an approach to visualize them to get a direct feeling of the information, and then we can pick an appropriate direction to tackle it. The effort comes in two folds: condense the data to a manageable magnitude and organize the information in a structured manner.

After applying the data cleaning techniques described in chapter 4.2.2, we utilize the features generated in chapter 5.2 to reduce the size of data points from half a million to about eighty thousands by grouping with tasks and ship numbers, where we still keep the individuality of each vehicle and task. The aspect we ignore for now is the details of each individual job at different time and locations, and we will take that into account in later stage of the analysis.

We select JMP as the platform to perform the analysis because of its availability and strong statistical, graphic and scripting abilities. The concept can be extended and carried out in another environment with similar capabilities.

The visualization tool we used first is the multivariate scatter plot, which is a scatterplot for each pair of the response variables displayed in a matrix arrangement. One point in a plot represents a record row in the database, and selecting one point in a plot will highlight the corresponding points of the same record row in all other plots. It helps us to visualize the relationship between metrics, and display the behavior of data in multiple dimensions. A multivariate scatter plot for the airline maintenance system is shown in Figure 5.2.1. Although it is a preliminary plot, it is an information rich plot, and needs some explanation.

From upper-left to lower-right, the descriptions of metrics are displayed along the diagonal axis, and the plot in the ith row and the jth column is the scatter plot for metric i and j with metric i on ordinate and metric j on abscissas, which we abbreviate as plot[i,j]. The lower-left off-diagonal plots are the mirror images of the upper-right off-diagonal plots, and do not contain extra information. The metrics listed in the matrix are: TotalTaskManHours (as TotalManHr in figure) depict total man-hours spent on a specific task for a ship; AverageTaskManHours (AvgManHr) presents the average man-hours required to finish the task; NumberOfOccurances (Occur) is the time of the task occurred in a ship; CostRank; Fleet (FleetNum); Ship (ShipNum), ATAChapterCode (ATA_CD); and Task (TaskId). Detail description of the metrics is discussed in Chapter 4.2.5.

For example, the plot[1,2] at the first row and the second column is the plot for TotalTaskManHours vs. AverageTaskManHours. The straight lines in the plot show that there is a linear relationship between the two metrics under certain condition. This can be verified by the concept of AverageTaskManHours, which is the product of TotalTaskManHours divided by NumberOfOccurances. When NumberOfOccurances is a constant, the points for TotalTaskManHours and AverageTaskManHours are on a straight line. With each unique value of the NumberOfOccurances, there is a line in this plot.

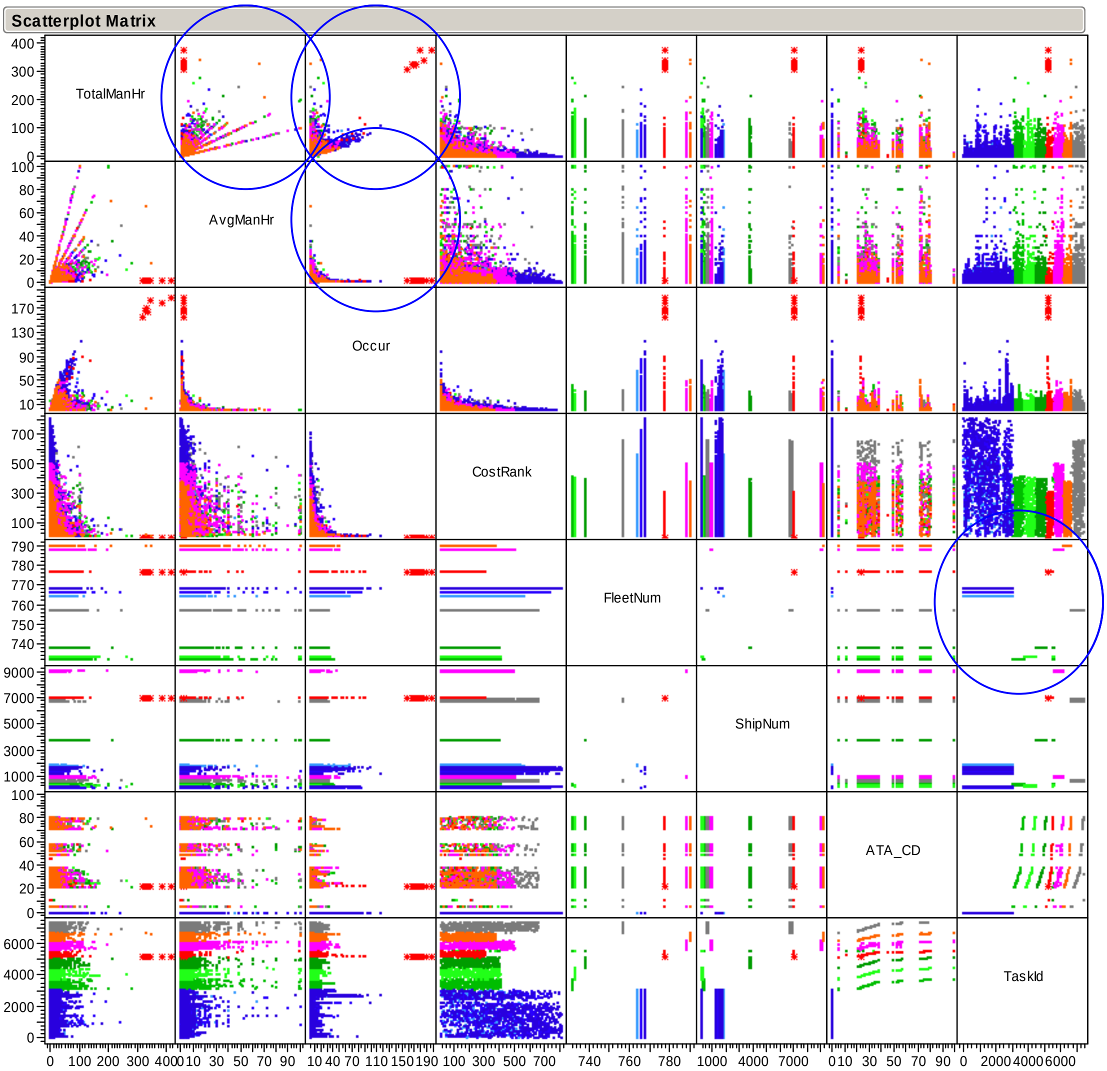



Figure 5.2.1 A Preliminary Multivariate Scatter Plot of the Airline Maintenance System

The plot[2,3] is the plot for AverageTaskManHours vs. NumberOfOccurances. We can see that these two metrics are in an inverse relationship, where longer tasks occurs less often and frequent tasks are usually shorter. One may notice that there are a few outliers in the plot[1,3] (NumberOfOccurances vs. TotalTaskManHours), which reveals some tasks with extremely high frequency. And those points are lined up to form a straight vertical line in the plots at the columns of the fleets and tasks, and they collapse into one point in plot[5,8], which means a single task in a particular fleet. Checking with the database, it is found that the task is IFE (In-Flight Entertainment system) on Boeing 777, which has occurrence of over 150 times in each of the Boeing 777 during the eight months period. With the AverageTaskManHours around 2 hours, this task consumes many maintenance man-hours. The reason for this task occurs in such an abnormal high frequency on Boeing 777 will be discussed in later section.

This plot is displayed at a certain degree of abstraction with individual jobs grouped into task, and fleets are colorized for better visualization. For example, points associated with Boeing 777 are in red, and those with Boeing 737-300 are in light green. As one can see, it is hard to obtain useful knowledge directly from this plot, since the points are all over the place, and no other significant trend can be observed besides those discussed above. We need some technique to further investigate the data.

With the hierarchical approach described in chapter 4.2.5, we establish multiple levels of hierarchy. The top level depicts the system effectiveness at the system level, and describes how well the system functions. These considerations include availability, maintainability, reliability and cost.  The second level represents the vehicle level, and it shows the behavior of a group of vehicles (fleet) or an individual vehicle, called ship. The third level represents the component level, and it shows the performance of each component according to ATA chapter codes and tasks. The system availability is represented by TotalTaskManHours, which is a feature generated in the feature exploration step. Assuming that the aircraft is available if it is not serviced in a repair station, a higher value of TotalTaskManHours implies a lower aircraft availability. The system maintainability is represented by the variable AvgTaskManHours. In this context, it is assumed that a system has a low maintainability value if it requires a longer service time to repair. Reliability in this thesis research is represented by the occurrence of a task in a given period. A system assumes a low reliability value if it requires frequent repairing.

With this hierarchy, we organize the information in Figure 5.2.1, and create a more structured presentation in Figure 5.2.2. The metrics of system effectiveness are grouped at the upper left diagonal, and the level in hierarchy decreases along the diagonal to the lower right corner.

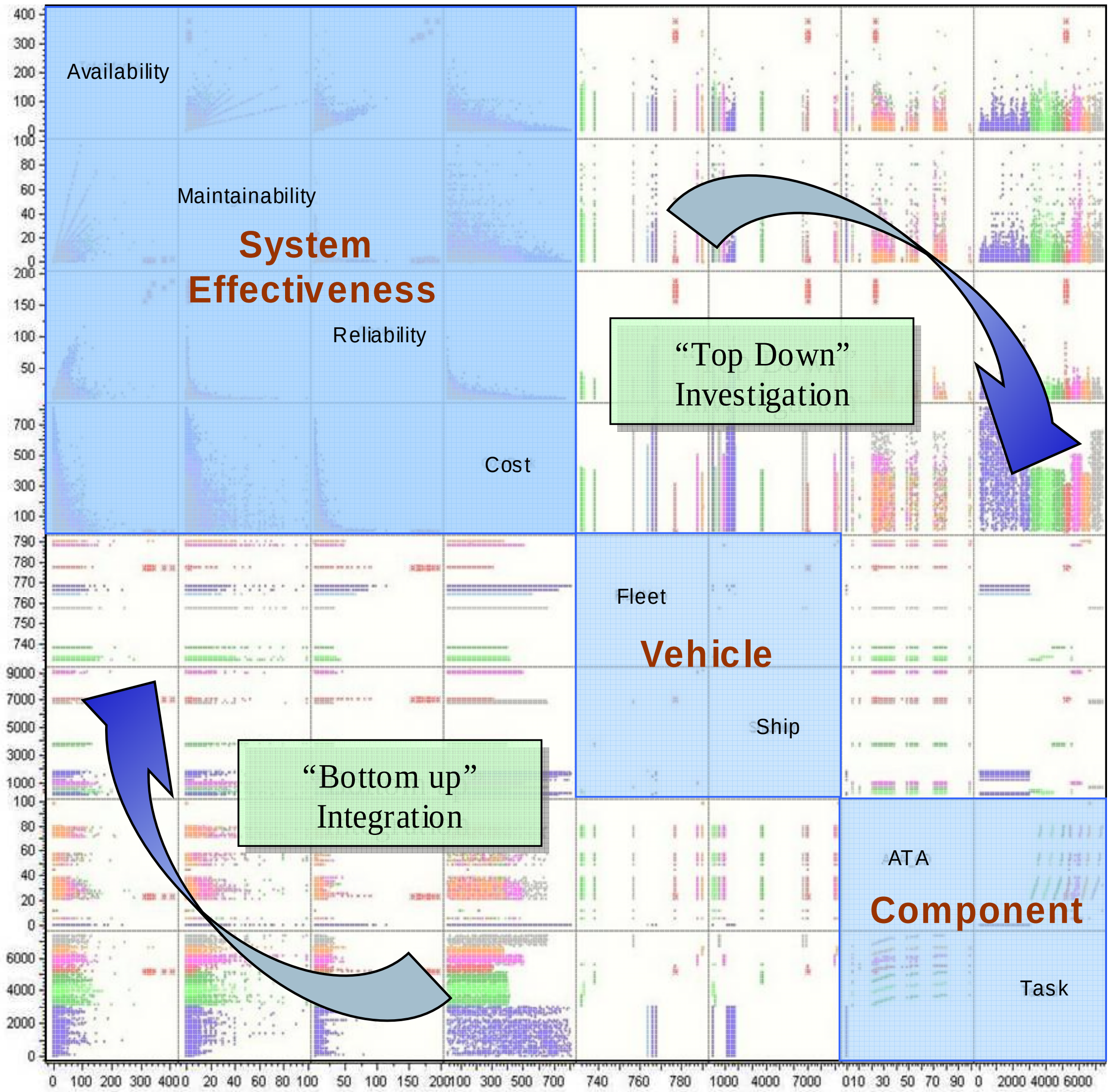



Figure 5.2.2 A Hierarchical Multivariate Scatter Plot of the Airline Maintenance System

As discussed in chapter 4.2.5, a bottom up approach is taken to integrate the lower level systems into a higher-level system using a system composition process so as to obtain a holistic view of the overall situation. Subsequently, a top-down approach is taken to decompose the complex system into smaller subsystems to reduce the scope of the analysis by investigating each of the sub level systems in an attempt to find potential causes of an identified abnormality.

5.2.1.2 Bottom-up Integration

We first use the bottom-up integration to compare performance between fleets, where all the information regarding a fleet is summed up into one record, and integrated as one point in a scatter plot. By this bottom-up approach, we reduce the number of points from eighty thousands in the previous plots to ten. The new multivariate scatter plot matrix at fleet level is illustrated in Figure 5.2.3.

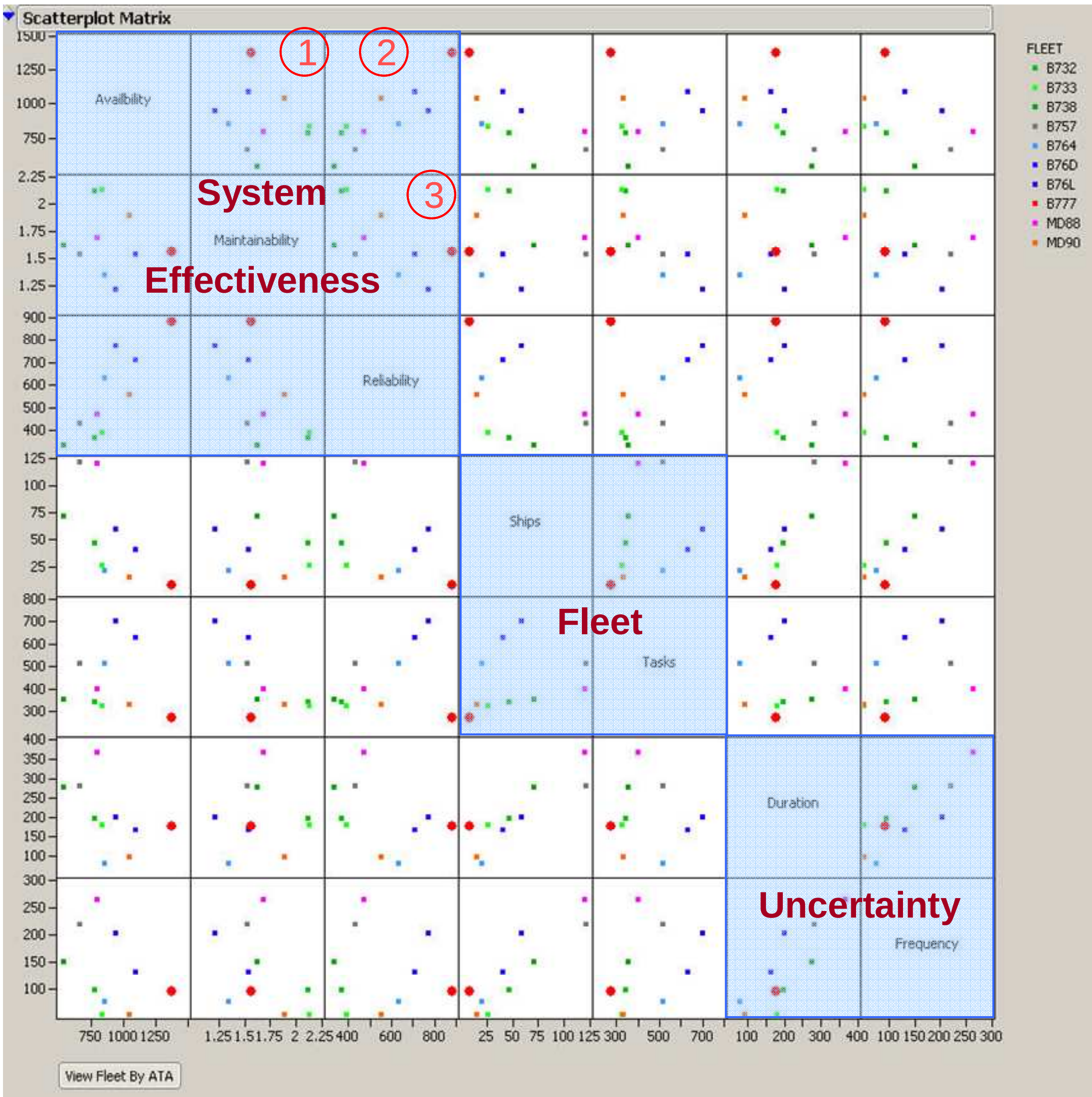



Figure 5.2.3 Fleet Level Multivariate Scatter Plot

Each point in the above matrix represents a fleet, and it is displayed in a unique color to ease the comparison between fleets. With the integration, we have a significantly simplified visualization, and the hierarchy is still in place from the system effectiveness to fleet information. The information about vehicles and components are ignored at this stage, since we are focused on the fleet level. We will examine the details later at proper stage. The uncertainty to the integration, which is measured by standard deviations of the metrics, is also displayed in the lower right of the matrix for analysis.

The relationship between the system effectiveness metrics can also be displayed as a 3D interactive plot in JMP (Figure 5.2.4), and users can interact with the plot by rotate the axes or changing ranges for a better understanding of the information.

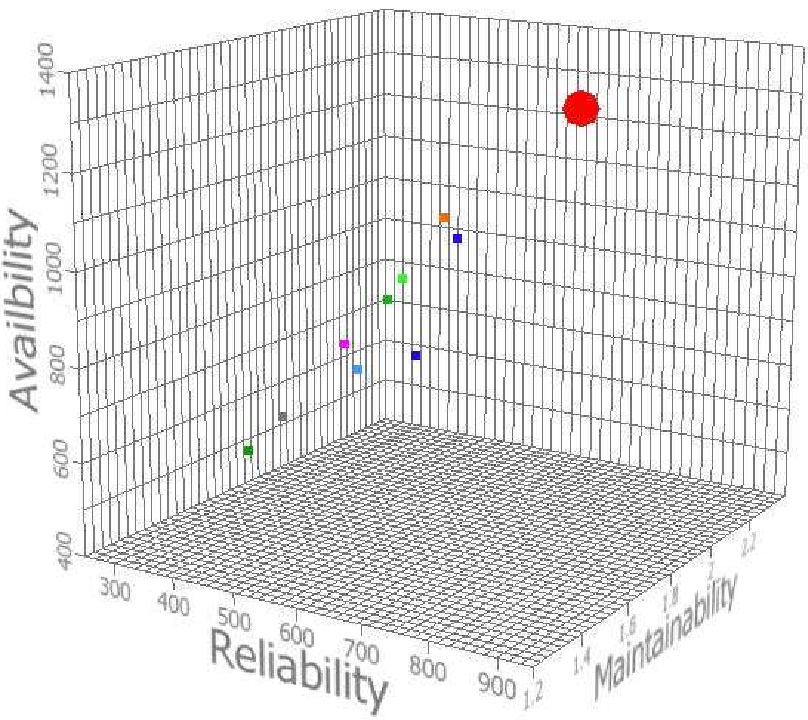



Figure 5.2.4 System Effectiveness Metrics in 3D Plot

5.2.1.3 Top-down Analysis - Reliability

With the support of the JMP Scripting Language (JSL), we create a top-down investigation environment in JMP. JMP data tables are created based on the information found in the database, and a visual investigation process is presented in this thesis. The environment interacts with the users, and guides them through the investigation. We use an example in the context of aircraft maintenance system, and the concept can be used in other system analysis with modification.

We first run a JMP script file, called TopdownAnalysis-acMaint.JSL, and it will initialize the JMP environment, load data tables, and display a JMP window (Figure 5.2.5) with fleet level multivariate scatter plot matrix similar to the previous plot. There is a button, titled "View Fleet Detail", at the bottom of the window, which enables the link between the current data table and other data tables for top-down analysis.

In this example, we focus on the fleet with the lowest availability, which has the highest value in TotalTaskManHours/Ship, and try to figure out what is the cause of the problem. In plot[1,2], we click and highlight the red point, which is the highest point in the ordinate (TotalManHr, a measure of availability), and the points corresponding to the same fleet in other plots are all highlighted. We move the cursor over the point of interest and a popup label appears identifying that the fleet is comprised of Boeing 777-200ERs. The red point is in the middle of the abscissas (AvgManHr, a measure of maintainability) in plot[1,2], which means the aircraft in this fleet does not take too long to fix in average. In plot[1,3], the highlighted point is at the upper right corner with the highest value in abscissas (Occur, a measure of reliability), which means the aircraft frequently need maintenance, and some components are not reliable.

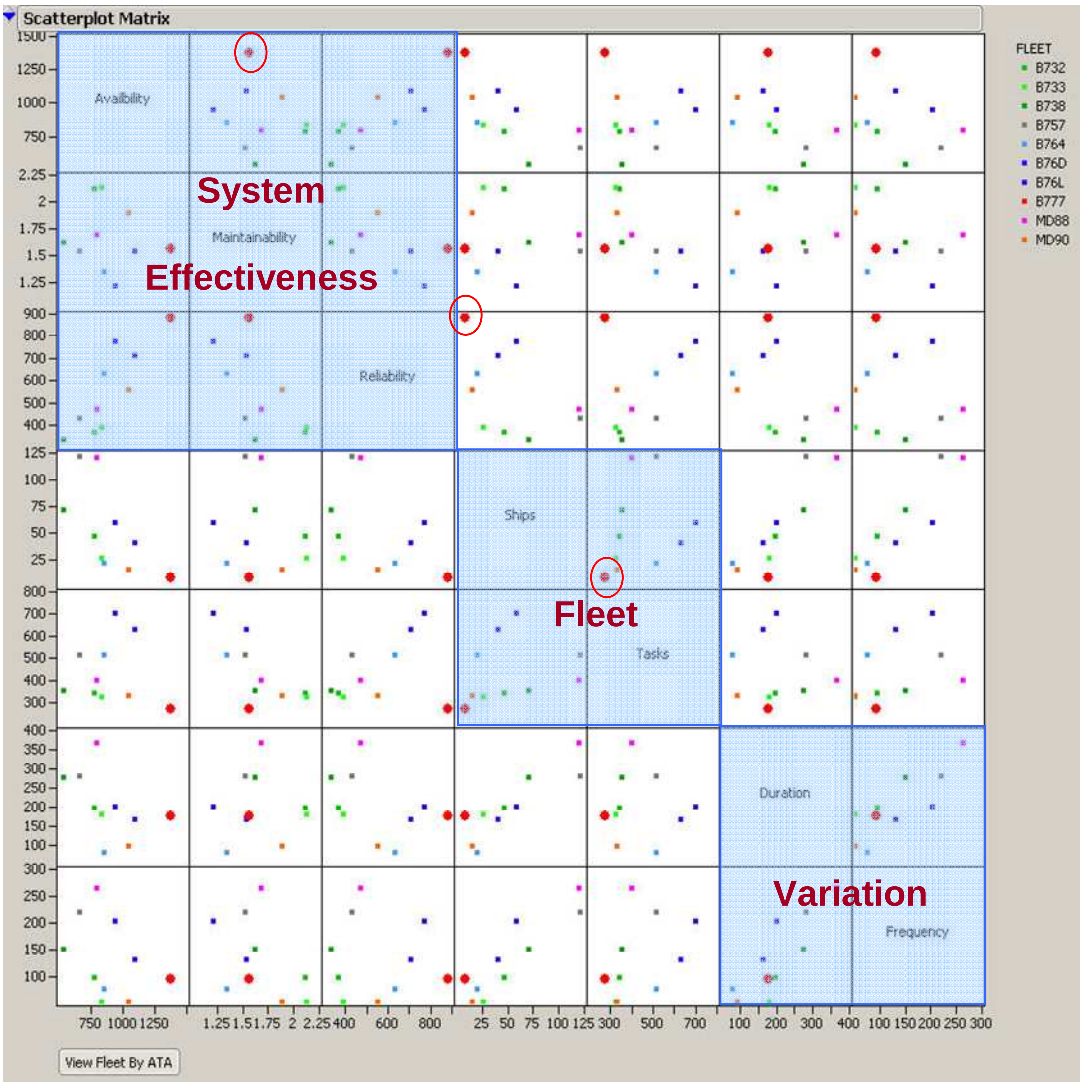



Figure 5.2.5 Top-down Analysis at Fleet Level

Figure 5.2.5 reveals the primary reason for low availability – low reliability in some components. We need to go to the component level to find out which components cause the problem, and we click on the “View Fleet Detail” button to bring the component level information for the selected fleet, which is Boeing 777-200ER, see Figure 5.2.6.

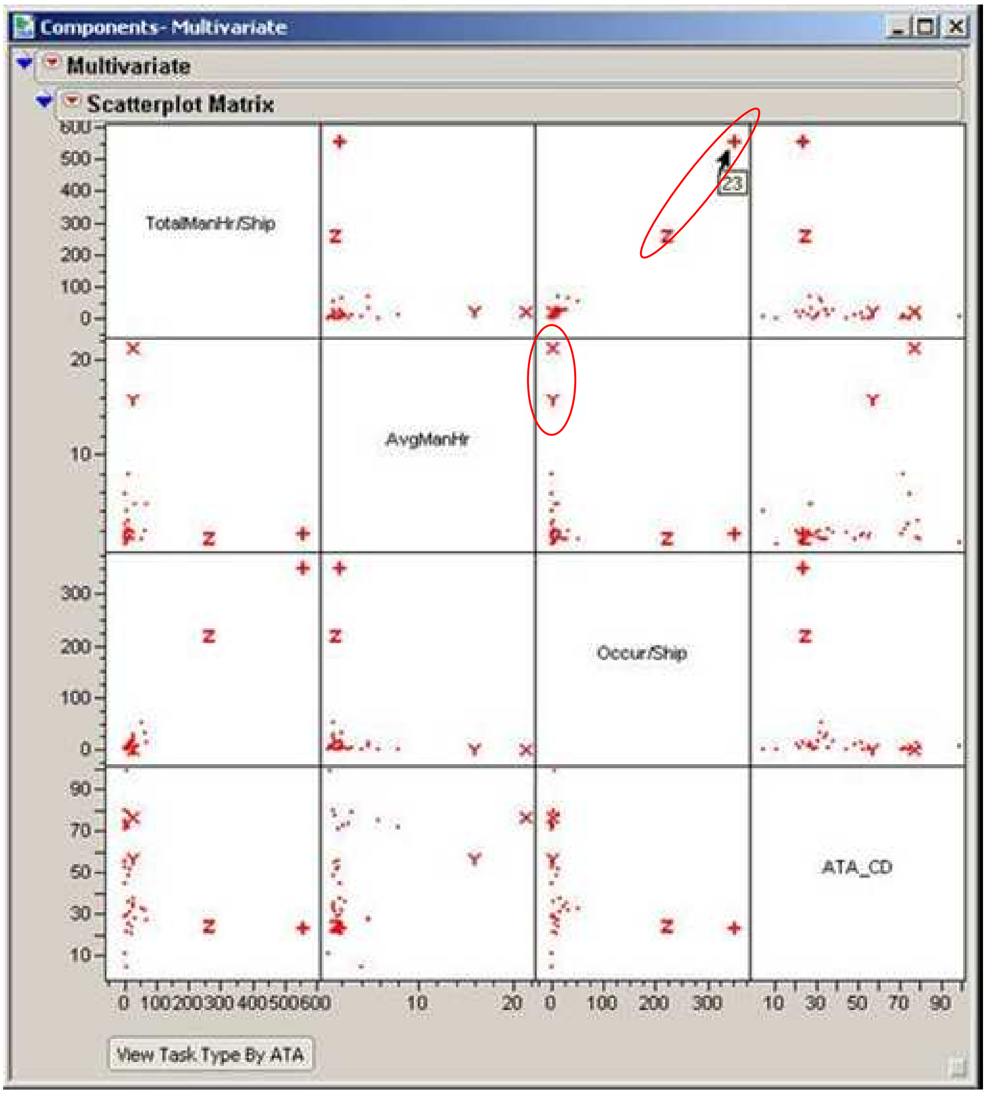



Figure 5.2.6 Top-down Analysis at Component Level

The above figure shows all the service components for Boeing 777-200ER, and each point in a plot represents a component by ATA chapter code. We use system effectiveness metrics to evaluate the components. As highlighted in the figure, there are four components stand out from others with their high values in one or two dimensions. The two points marked as “+” and “z” are high on TotalManHr/Ship and Occur/Ship, which means that they have low availability and low reliability; The other two points marked as “x” and “y” are high on AvgManHr, which means that they have low maintainability.

Since we are concerned the reliability of the system based on the observation of system level effectiveness in Figure 5.2.6, we first take a look at the component with lowest reliability, which is the point “+”, and will come back to visit other highlighted components in later section. We move the cursor on the point, and the popup note shows that it has ATA chapter code of 23, which is the Communication component (Appendix A). We select this point and click the button “View Task By ATA”, and the top-down environment brings us to the next level of the system down in the hierarchy, which is the task level. In Figure 5.2.7, we have two tasks stand out of the others with high Occur/Ship, which are marked as “+” and “x”. Hover the cursor on “+”, it shows information regarding this task, i.e. “B777, 23, 5188, IFE, 350.39, 175.12, 2”. It is a task on Boeing 777 with ATA chapter code 23, task ID is 5188, task description is “IFE” (In-Flight Entertainment). The system effectiveness measures for this task are: TotalManHr/Ship = 350.39 hours, Occur/Ship = 175.12, and AvgManHr = 2 hours. It means that this task requires on the average of 2 man-hours per service, and it occurred 175.12 times per ship (averaged among ships) and took a total of 350.39 man-hours in the eight-month record period.

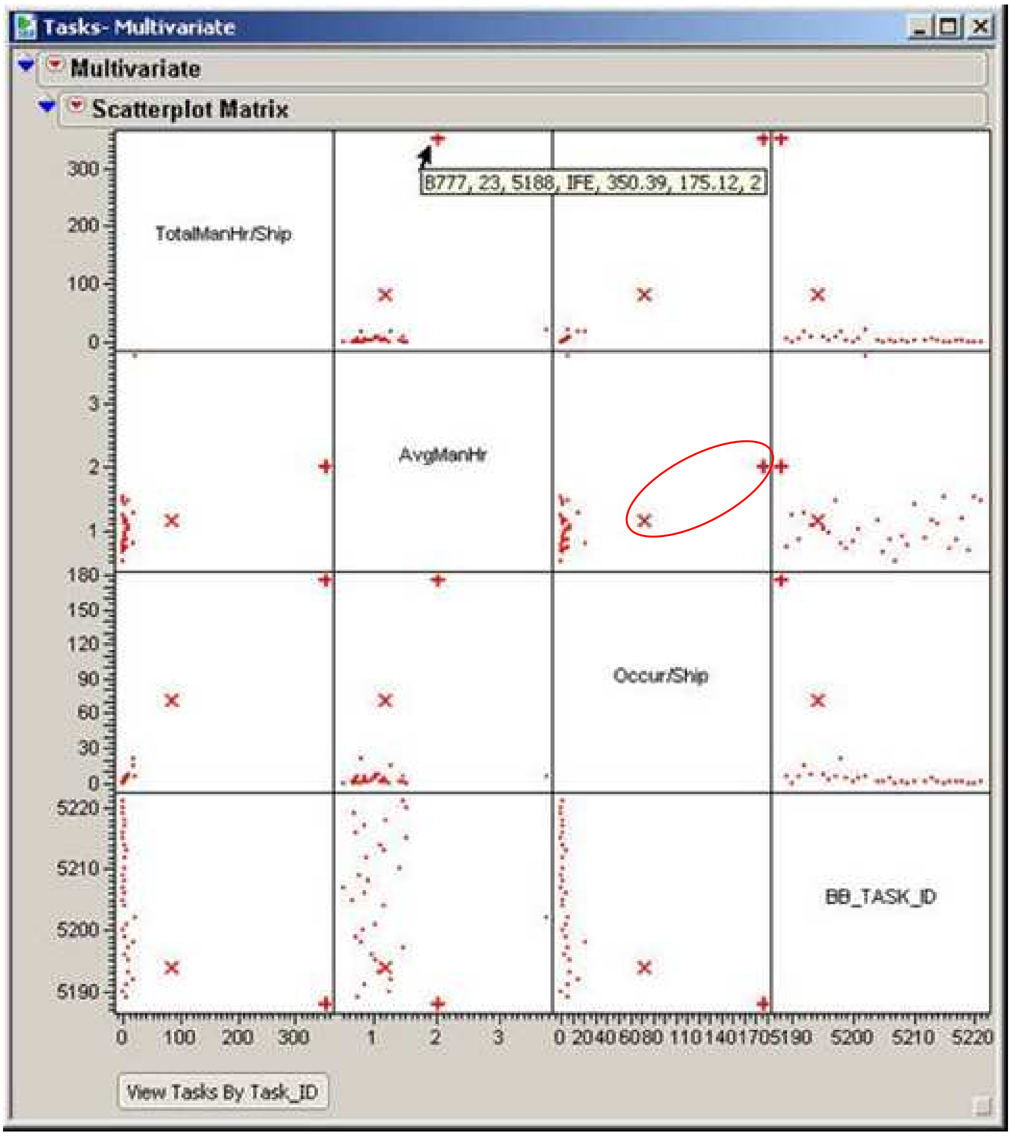



Figure 5.2.7 Top-down Analysis at Task Level

Based on the above observation, we can say that the IFE system on the airline's Boeing 777-200ER a very unreliable system, since it requires services of over twenty times each month on average. This conclusion also confirms our early observation in the preliminary multivariate scatter plot in Figure 5.2.1. To reduce the uncertainty of our investigation, we can go one step farther to check each individual jobs of this task (IFE) for each of the ships in the fleet. We select the "+" point and click "View Jobs By Task ID" button, the environment brings us down to the individual job level, Figure 5.2.8.

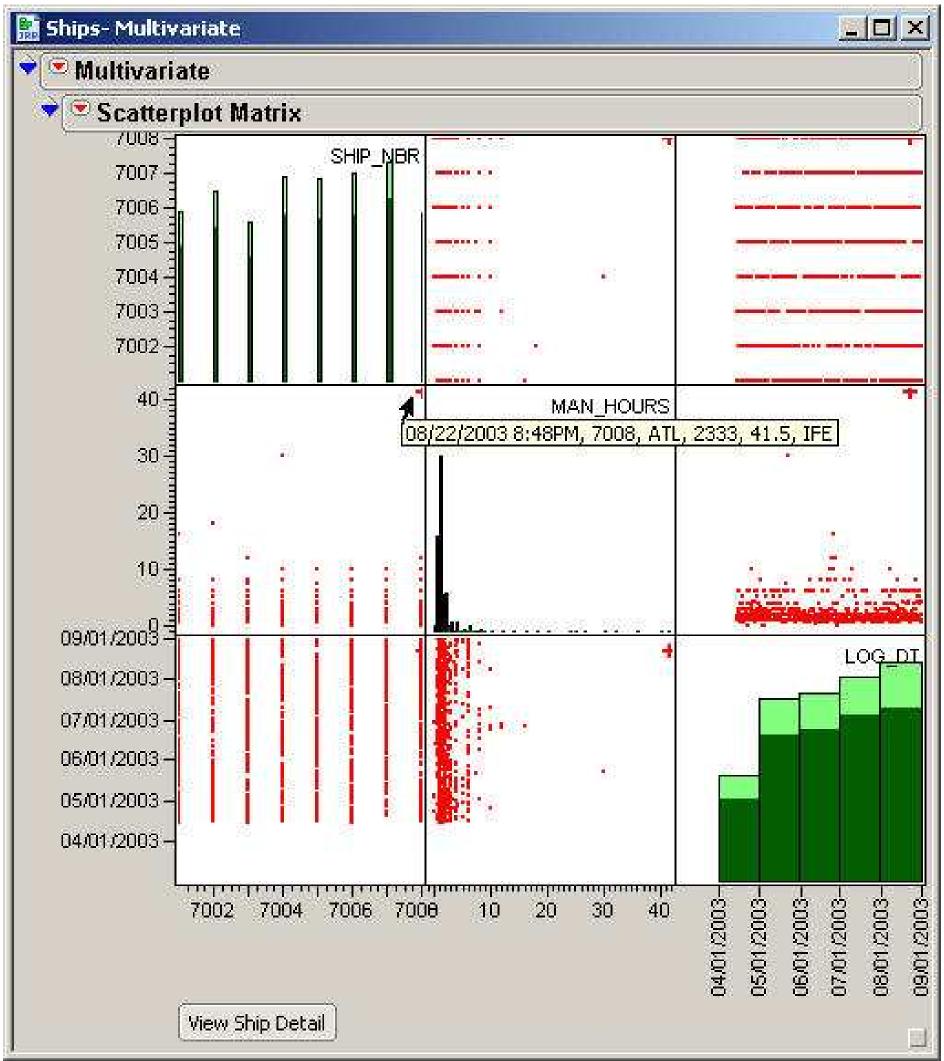


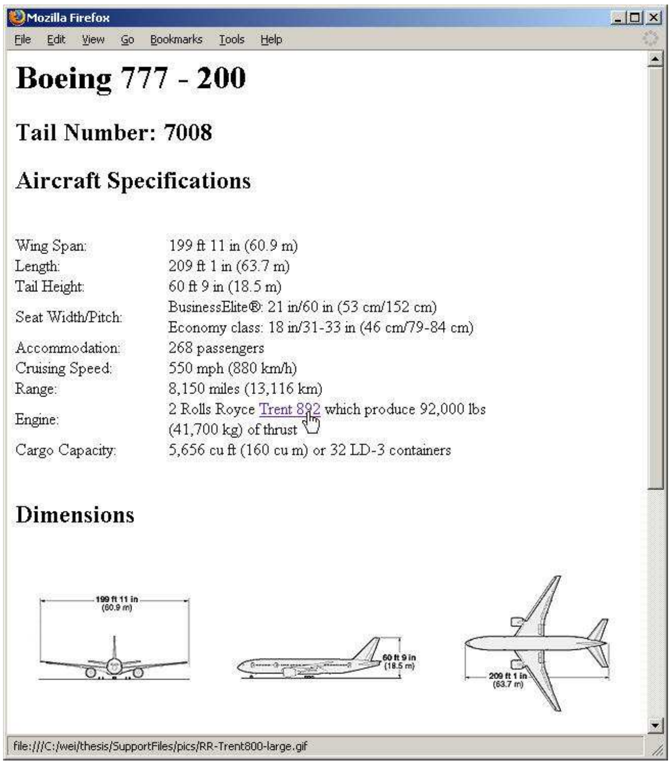

Figure 5.2.8 Top-down Analysis at Job Level

In this step, each point in the plot represents a single service done as IFE. We use ship number (SHIP_NBR) to identify the individual aircraft, and we also display the duration (MAN_HOURS) and the date (LOG_DT) of the individual services. A horizontal histogram of each metric is also shown in the diagonal cell, which depicts the distribution of jobs. We can see the IFE tasks are quite evenly distributed among all the ships, and across the time horizon. The month of April has lower occurrence because it is a half month record. Therefore, we can conclude that there is a problem in the IFE system of the Boeing 777-200ER aircraft, and designs could be improved to have a better or more reliable system.

After discussed with the experts in the airline service department, we presume the possible reasons for such a high IFE repair rate on its Boeing 777 fleet:

- During flight, the rapid changes in altitude may cause sudden pressure and humidity change, which makes the IFE system unstable;
- Time to market pressure forces Boeing to use off-the-shelf product which has less consideration on environment changes, and thus less reliable;
- Boeing 777 in this airline was relatively new at that time, and it takes some time to get stabilized

Although it is a general trend that an IFE service takes around two hours to finish, which can be confirmed in the distribution of MAN_HOURS, some jobs took longer amounts of time. For example, one service (the "+" point) on ship number 7008 on August 22nd 2003 took 41.5 man-hours in Atlanta, as shown in the label. We can select the point, and click "View Ship Detail" button to see the details of the specific aircraft. A webpage will popup, and display the related information of this aircraft, Figure 5.2.9. It is worthy to mention, the webpage is made by author himself for demonstration purpose, and any type of information can be linked to the webpage upon availability. For example, the details of the Rolls Royce Trent 892 engine on this aircraft can be displayed by clicking on the link.



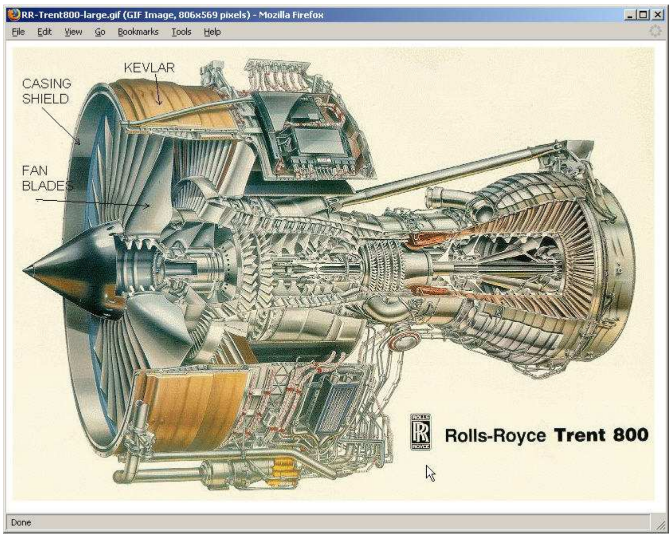




Figure 5.2.9 Top-down analysis - Ship Detail

5.2.1.4 Top-down Analysis – Maintainability

As revealed in previous section, we observed some components with low maintainability, which are point “x” and “y” in Figure 5.2.6. We ignored them at that time since we were focused on system reliability. In this section, we will take a look at the maintainability issues on Boeing 777.

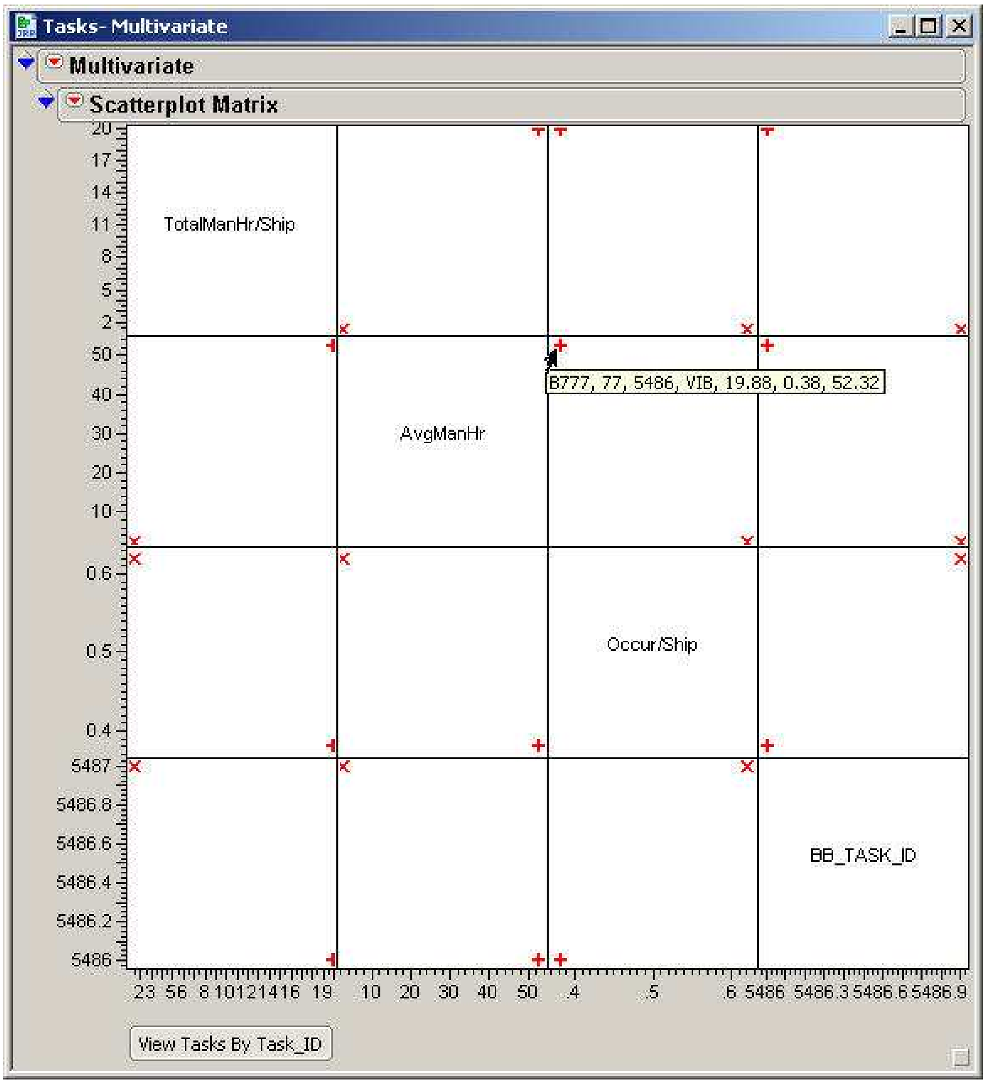



Figure 5.2.10 Component Analysis – Engine Indicating

In Figure 5.2.6, hover over the point "x", we found out the ATA chapter code for this component is 77, which is Engine Indicating. Select the point and click "View Task By ATA", the environment shows the Figure 5.2.10, where we can see two points. The task, whose task is VIB (Vibration) with task ID of 5486, at the point "+" has a very high AvgManHr at 52.32 hours for each service, while it is not a frequent occurrence (Occur/Ship is 0.38). The other task under ATA chapter at the point "x" has a low AvgManHr (2.23 hours) and low Occur/Ship (0.62), and it is an ordinary task. Select the "+" point and click "View Jobs By Task_ID" button, we can review the individual VIB jobs in Figure 5.2.11.

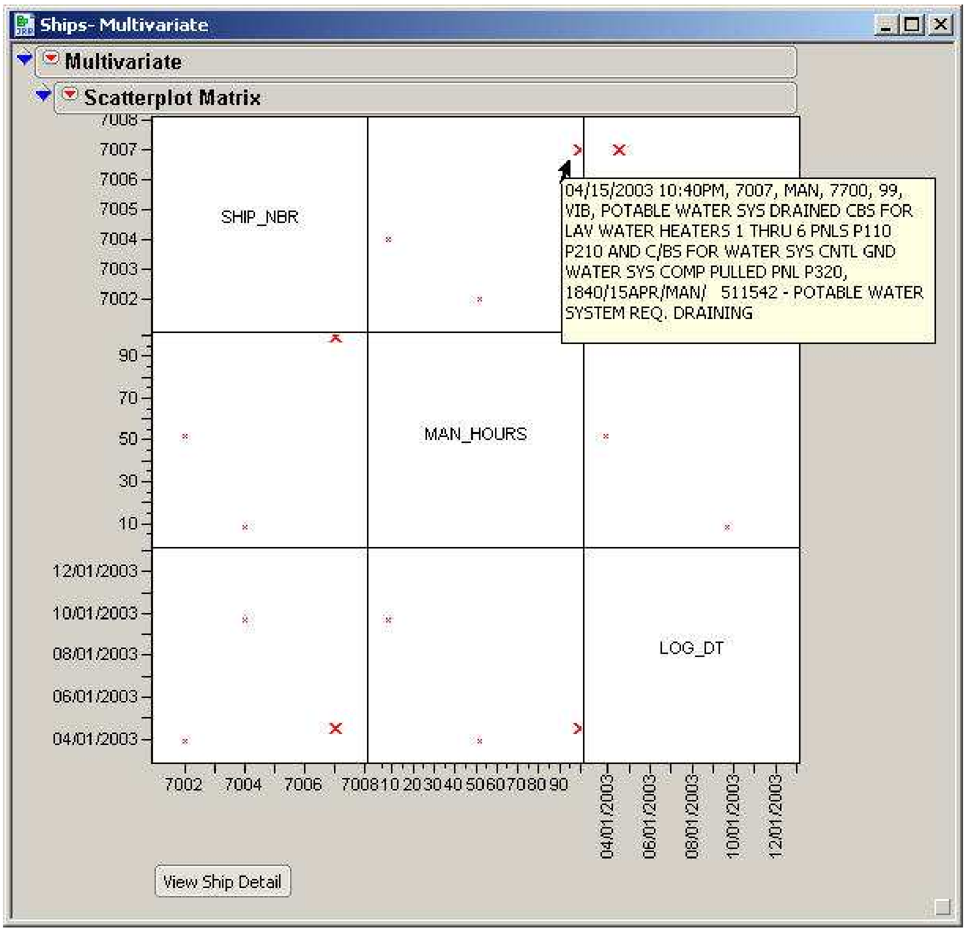



Figure 5.2.11 Task Analysis – VIB

There are only three occurrence of VIB, however, most of them required a long time from 10 man-hours to 99 man-hours. Among them, the service on April 15th 2003 for ship 7007 took 99 man-hours to finish. The details of inspection report and care report are shown in the label.

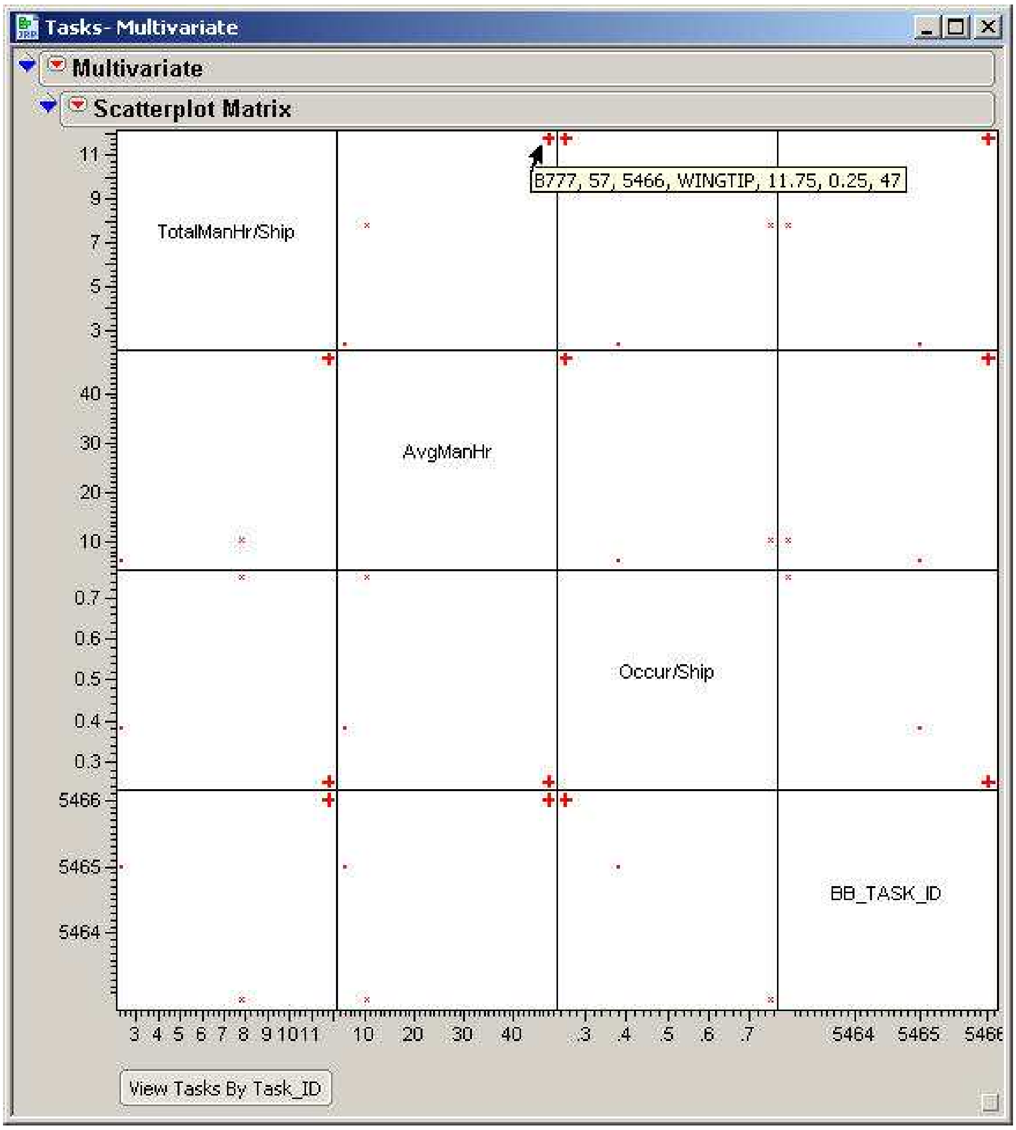



Figure 5.2.12 Component Analysis – Wings

For the point "y" in Figure 5.2.6, it is ATA chapter code 57 (Wings), and there are three tasks under this category shown in Figure 5.2.12. Among them, the task Wing Tip with task ID of 5466 has a high AvgManHr at 47 hours for each service, while it does not occur often (Occur/Ship is 0.25). Other tasks require much lower service man-hours. The further task analysis in Figure 5.2.12 shows two services encountered, and one of them took 90 man-hours on August 15th 2003 for ship 7004.

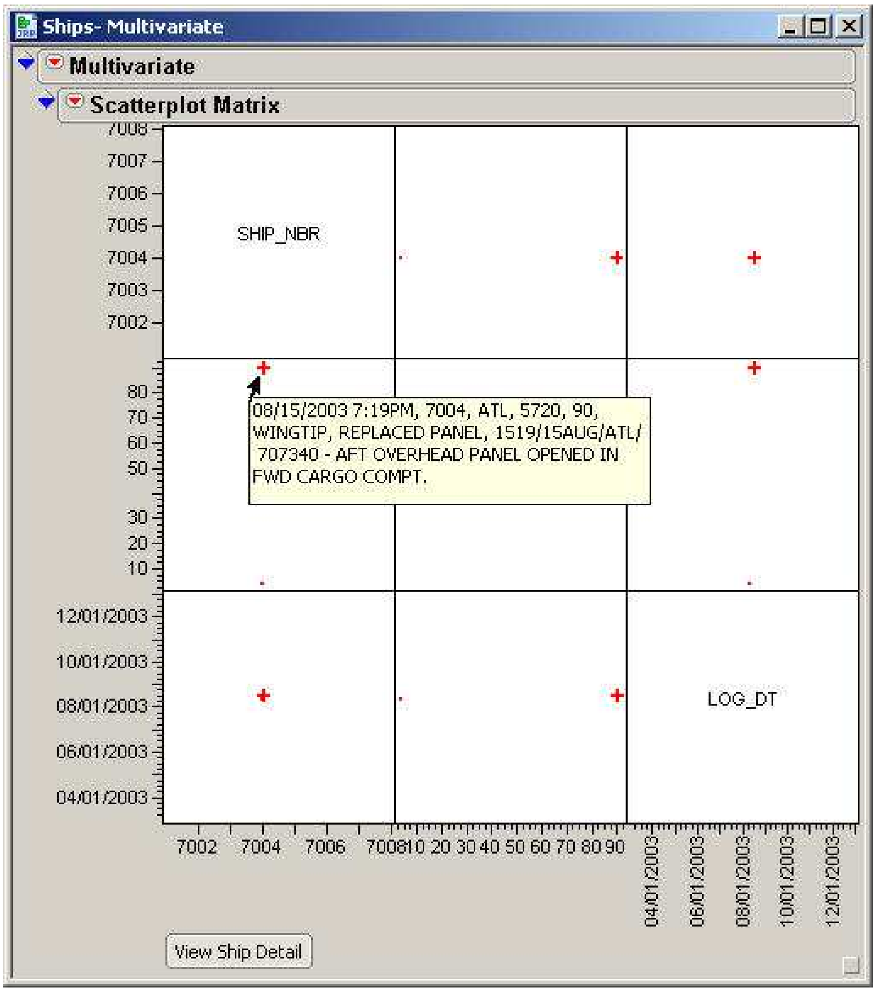



Figure 5.2.13 Task Analysis – Wing Tip

The impression we have from the above investigation are:

- Although it rarely happens, the task VIB and Wing Tip are hard to fix, which depicts low maintainability in design.
- The maintainability on the related parts could be improved in aircraft design.

### 5.2.2 System Analysis on Airline Operations

In this section, we evaluate the air transport system on-time performance to illustrate the usability of the multidimensional hierarchical system analysis in the operations field. The database we use includes all the flights in the major domestic commercial airlines in January 2005 with about 0.6 million records.

The database is obtained from the Bureau of Transportation Statistics [BTS 2005] The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics tracks the on-time performance of domestic flights operated by large air carriers. Summary information, on the number of on-time, delayed, canceled and diverted flights appears in the DOT's monthly Air Travel Consumer Report, which is published on its website and made available to the public since June 2003. In this website, BTS provides some basic summary statistics, such as summation, average, count, and so on. Our approach establishes system evaluation metrics, and reveals the hierarchical relationships between the metrics in a user-friendly visualization environment. Observations can be better made to support decision making.

The sample data are used to evaluate the on-time performance of the air transportation system by different categories at different levels. A system hierarchy is illustrated in Figure 5.2.14. The national air transportation system includes carriers, airports, and flights operated by carriers transport passengers and goods between airports. Each flight starts at a departure time and airport and ends at an arrival time and airport. The flight information contains flight number, departure/arrival times, delay times and causes, air time, flight distance, etc.

Delay causes are categorized as carrier delays, when the cause was due to circumstances within the airline's control (e.g., maintenance or crew problems, aircraft cleaning, baggage loading, fueling, etc.); weather delays, when significant meteorological conditions (actual or forecasted) that, in the judgment of the carrier, delays or prevents the operation of a flight (e.g., tornado, blizzard, hurricane, etc.); National Airspace System (NAS) delays, which are attributable to the national aviation system and they refer to a broad set of conditions — non-extreme weather conditions, airport operations, heavy traffic volume, air traffic control, etc.; security related delays, which are caused by an evacuation of a terminal or concourse, re-boarding of aircraft because of a security breach, inoperative screening equipment and/or long lines in excess of 29 minutes at screening areas; and late aircraft delays, where a previous flight with the same aircraft arrived late, causing the present flight to depart late.

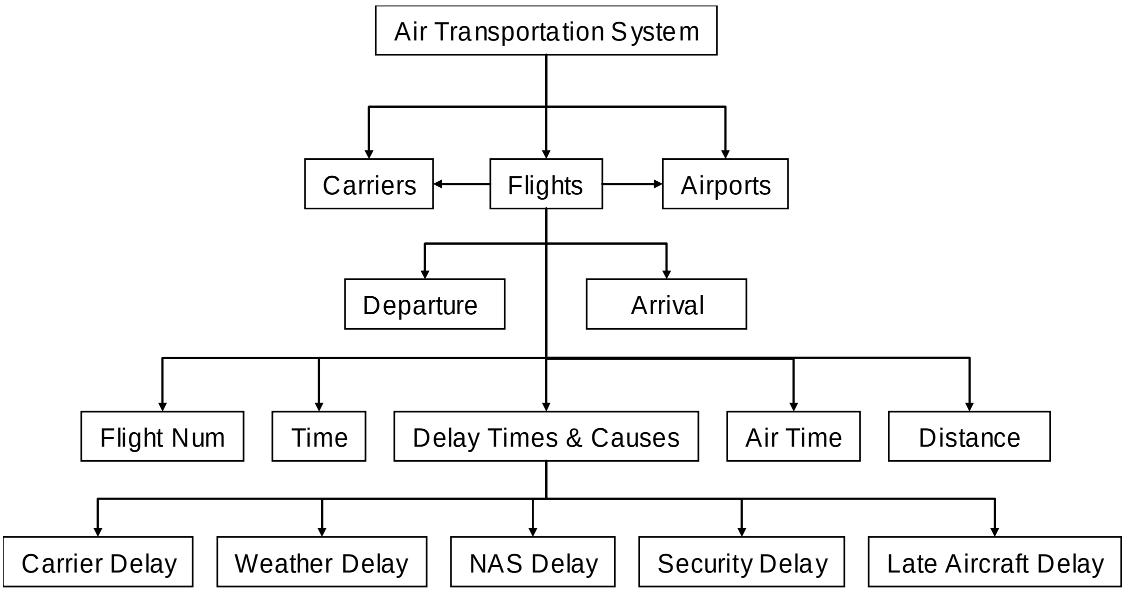



Figure 5.2.14 Sample Hierarchy in Airline On-time Performance

5.2.2.1 Preliminary System Analysis

To simplify the illustration, we do not include cancelled/diverted flights, and only show the flights delayed over 30 minutes in either departure or arrival to put our focus on delayed flights. About 100K points are colored by airlines in the multivariate scatter plots. The metrics shown are in two categories:

Flight information: day of month, day of week flight number, air time, distance, origin state Fips, which are two digit numbers to uniquely identify the states in the United States [FIPS 1994], Destination State Fips.

Delay Information: departure delay, arrival delay, carrier delay, weather delay, NAS delay, and late aircraft delay. See Figure 5.2.14.

With the illustrated air transportation system multivariate plot, we can carry out graphical analysis of air transportation system of systems:

Obtain overview of the system in various aspects. In the system demand analysis, we found early days of the month have higher demand, and the demand gradually reduced in the month of January 2005; and Monday, Wednesday, Thursday and Sunday have high traffic, Tuesday's traffic is lowest. Flight distance can be described with a Lognormal distribution. California, Illinois and Texas are the top 3 states in frequent air traffic. Overall observation at top level can result sub-level detailed analysis

Investigate the impacts of each type of delays to the on-time performance of the system. Counter-intuitively, late aircraft (36%), NAS (28%), and carrier (28%) delays are the primary reasons for flight delays, and weather (9%) and security delays are not as significant among all delay reasons.

Identification of errors in data. With this graphical multi-dimensional approach, it is very easy to spot certain erroneous data from huge amount of data. For example, some

Air Time data in Figure 5.2.15 have very high negative values and high positive values over 1440 minutes (which equals one day), and some Departure Delay in Figure 5.2.15 is made of very high negative values as well.

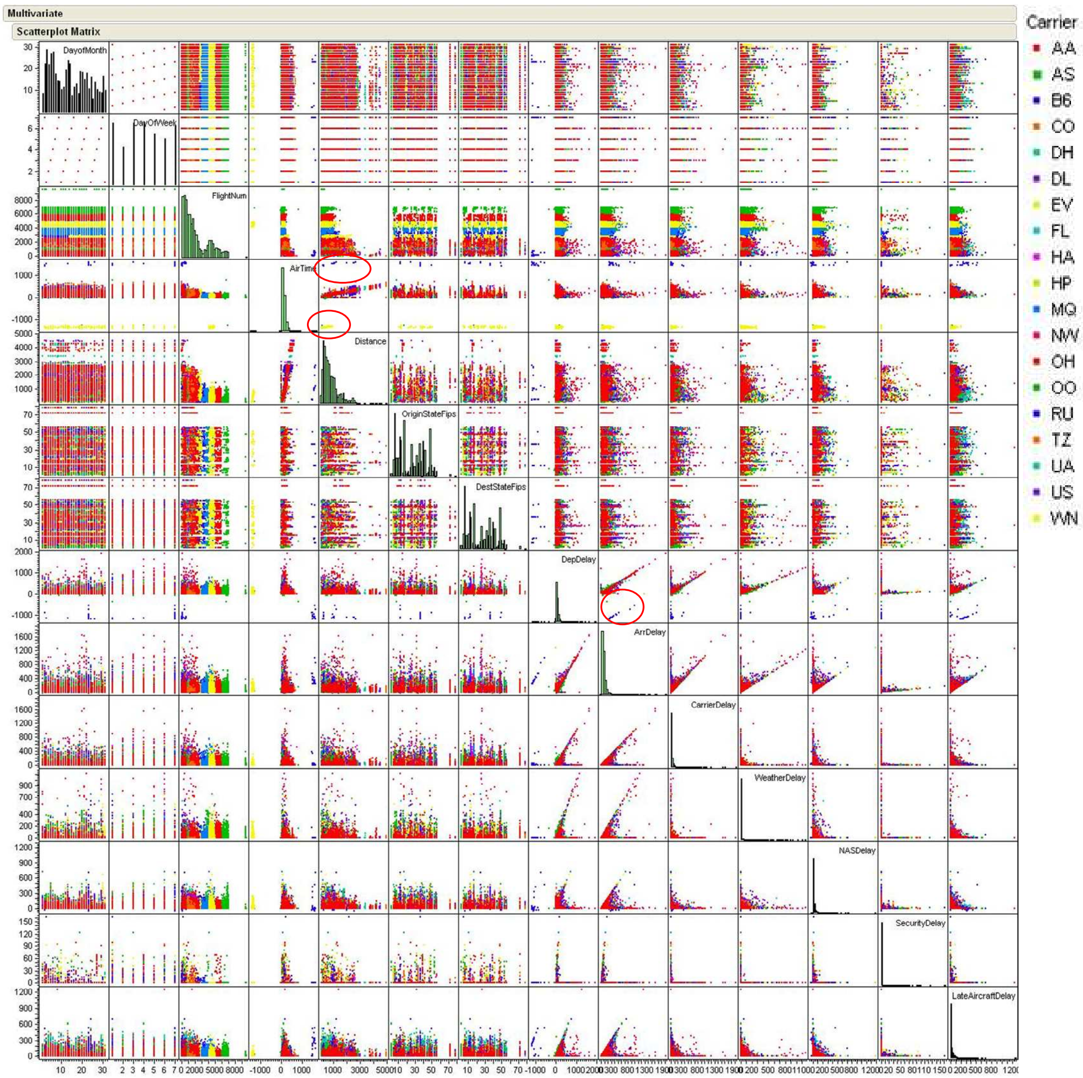



Figure 5.2.15 Preliminary Air Transportation System Flight Analysis

A potential reason of the incorrect calculation in the BTS database is time conversion when flights went across two days. Time zone may also have an impact if not handled correctly. With the above errors fixed, we now have new multivariate scatter plots in Figure 5.2.16.

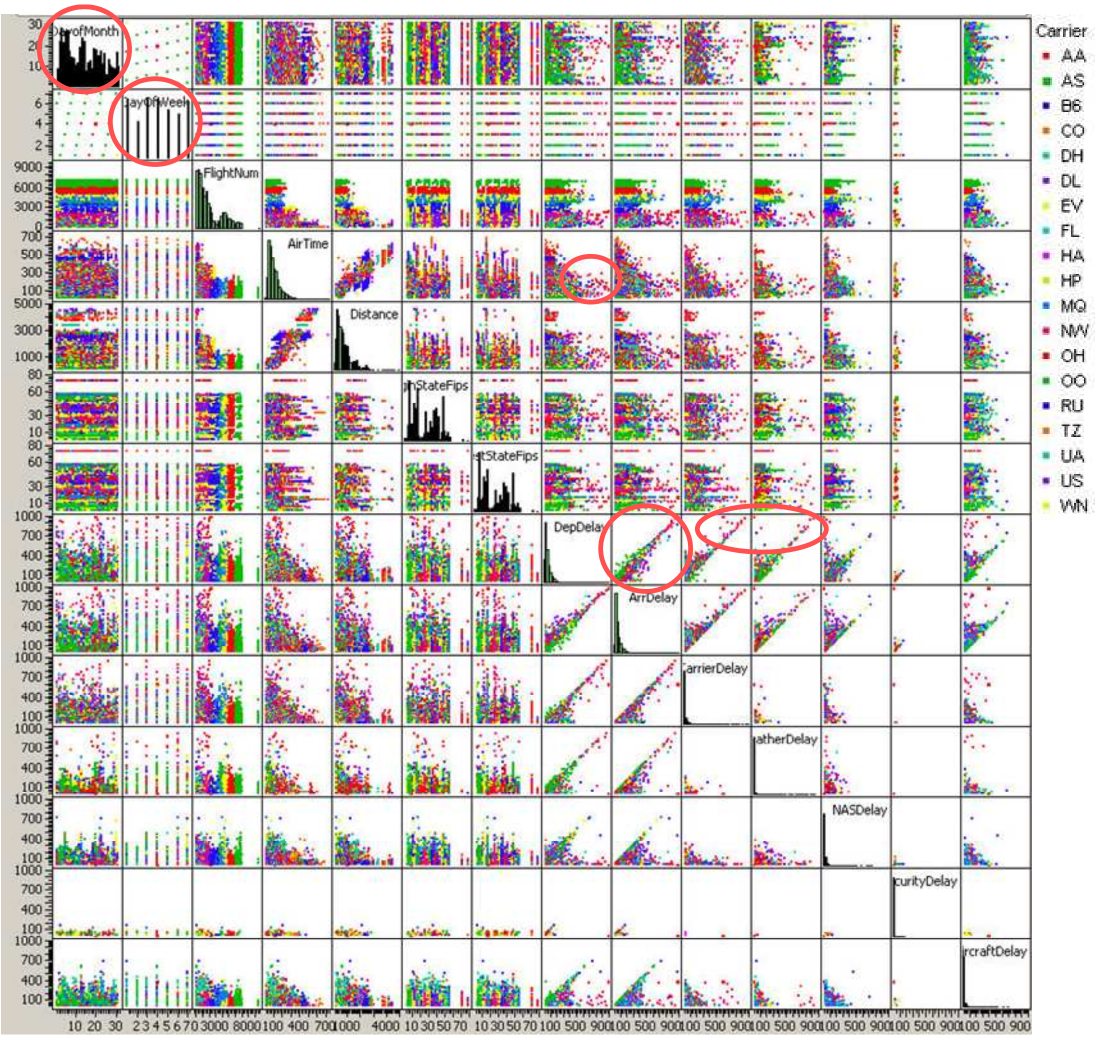



Figure 5.2.16 Preliminary Air Transportation System Flight Analysis with Error Fixed

Observations are made as follows:

- Higher demand appears in early of the month, then demand is slowly reducing
- Monday, Wednesday, Thursday and Sunday have high air traffic, Tuesday's traffic is lowest. Special ticket pricing strategy needs to be considered accordingly
- Long delays occur in shorter flights

- Arrival delays are correlated with departure delays, and often delayed longer
- Most of long delays are results of Carrier delay and Weather delay

5.2.2.2 Carrier Analysis

We are now focusing on comparing the carrier performance. Features are generated as system effectiveness metrics according to section 4.3. TotalCarrierDelay is to measure how much delay has occurred for all the aircraft in an airline in minutes; TotalDistanceKMile is to depict how many thousand miles an airline has flown in the investigation period; NumOfFlights is how many flights an airline has flown; AvgFlightCarrierDelay is a measure of carrier delay per flight; AvgKMileDelay is a measure of carrier delay per thousand miles; AvgFlightDistance is how many miles a carrier flies per flight.

With the bottom-up approach discussed in section 4.2.5, we integrate all the flights for each carrier into one record so that we can compare carrier performance easily. A multivariate plot of carrier system metrics is presented in Figure 5.2.17. In plot[4,5], we highlighted the worst performers in “x”, which have longer carrier delays over unit distance. The carriers are Comair Inc., Skywest Airlines Inc., Atlantic Southeast Airlines, and American Eagle Airlines Inc.. Although they have quite short flight total distances in plot[2,5], their total carrier delay is very high in plot[1,5]. Southwest Airlines Co. (shown as the point “+”) has the most number of flights in plot[3,4], and its average delays (AvgFlightCarrierDelay and AvgKMileDelay) are among the lowest, which presents a high on-time performance. The traditional big airlines, i.e. American Airlines Inc., Delta Air Lines Inc. and United Air Lines Inc., enlist the top fly distance, and their average delays per flight are at the relatively high end, which are two to three times more than Southwest Airline Co.

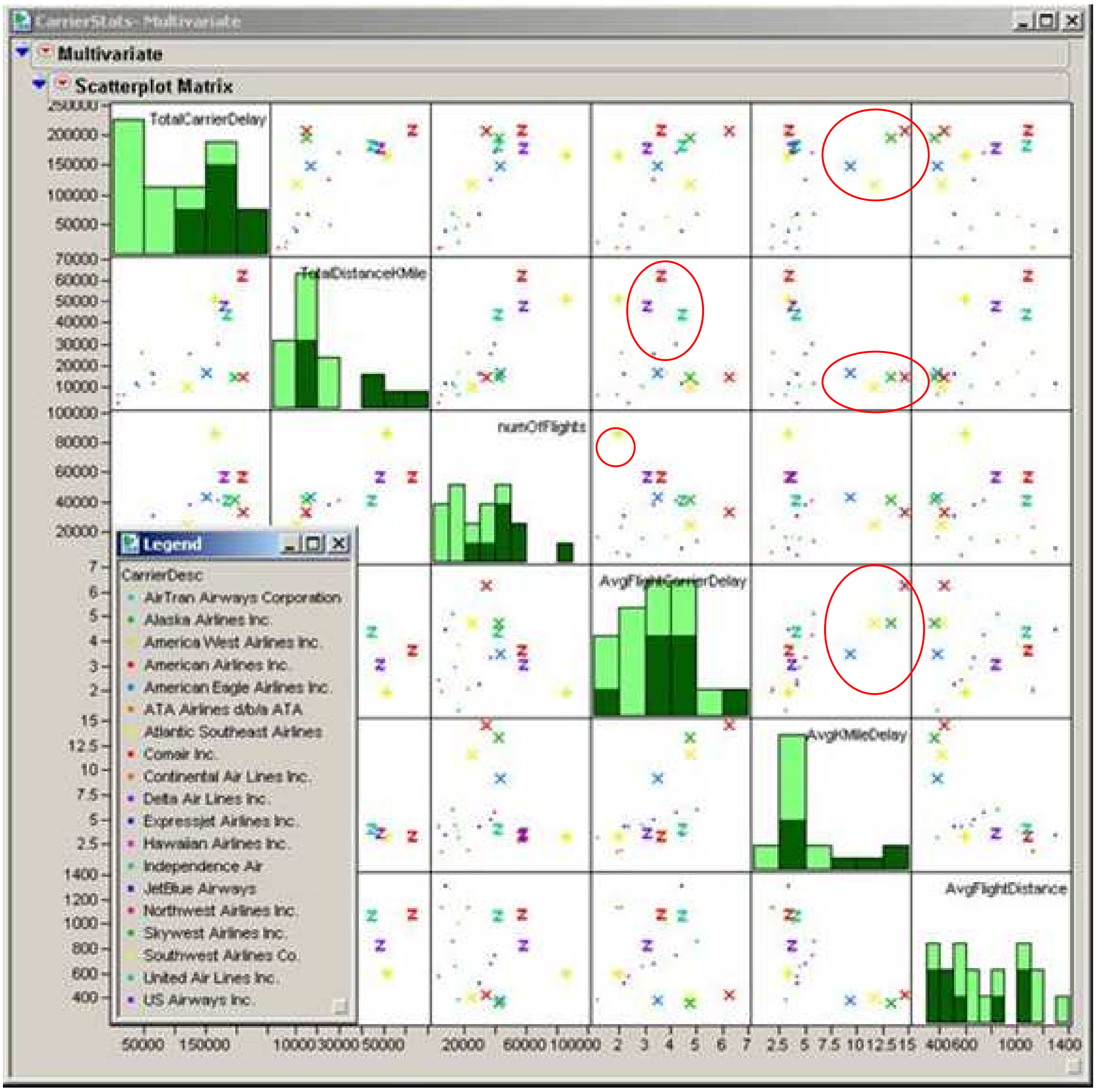



Figure 5.2.17 Carrier Operations Analysis

5.2.2.3 Airport Analysis

We apply the proposed graphical system analysis approach to examine the airports for on-time performance in the United States. Assuming takeoffs and landings are mostly symmetric, we only consider takeoff activities in the airports without losing generality. The delay metrics are normalized by dividing by number of takeoffs so that we are not comparing apples to oranges. The delay categories are discussed in the previous section. Although not all delay causes are directly linked to the airports, we can still use them to evaluate the airport operations.

We use the multivariate scatter matrix to visualize all 274 airports, where major carriers operate in the nation, Figure 5.2.18.

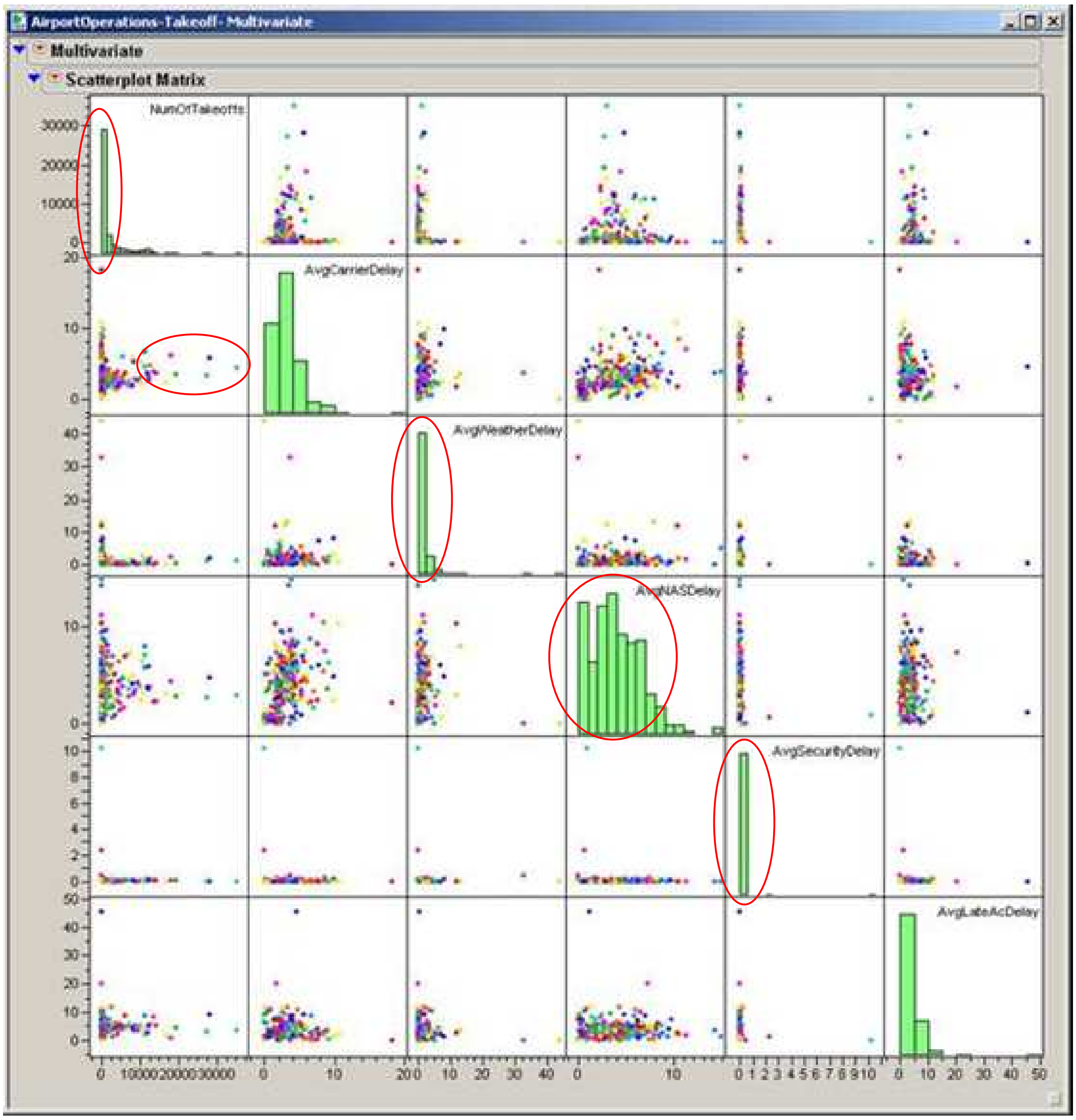



Figure 5.2.18 Delay Cause Analysis – All Major Domestic Airports

We have the following observations based on the data given in January 2005:

Majority of the airports have less than 1000 takeoffs each month, which is less than 33 takeoffs a day in plot[1,1]. Big airports, which have over 10000 takeoffs a month, do not delay flights longer than the small airports do in plot[1,*]. Long weather delays and security delays are only in certain regions in plot[3,3] and plot[5,5], while NAS delays are common in most airports in plot[4,4].

In the month of January 2005, a small airport (MQT) in Marquette, MI has the longest carrier delay at an average 18 minutes per takeoffs for total 88 takeoffs. Two airports, DRO in Durango, CO and CYS in Cheyenne, WY, have the highest average weather delay at average of 44 and 32 minutes per flight.Two airports, LNK in Lincoln, NE and MEI in Meridian, MS, have the highest NAS delay at average of 15 and 14 minutes per flight; The top 2 security delayed airports are AKN in King Salmon, AK and ADK in Adak Island, AK. The top late aircraft delayed airport is SCC in Deadhorse, AK. It seems the state of Alaska have more troubled airports than other states.

We are now focusing on the performance comparison of those big metropolitan airports. We take the top ten airports ranked by the number of takeoffs, where all the chosen airports have over 12000 takeoffs within a month. Among them, Atlanta Hartsfield-Jackson International Airport (ATL) had the most activities with 35,282 takeoffs in January 2005, which is average around 1,200 takeoffs per day and 50 takeoffs every minute if the flights are distributed evenly across the day. Within the 0.6 million total flights in the month, the top ten airports took one third of the activities (0.2 million), and ATL took about 6% of the national volume per number of takeoffs.

We first take a look at the average delay duration in different cause categories, Figure 5.2.19.

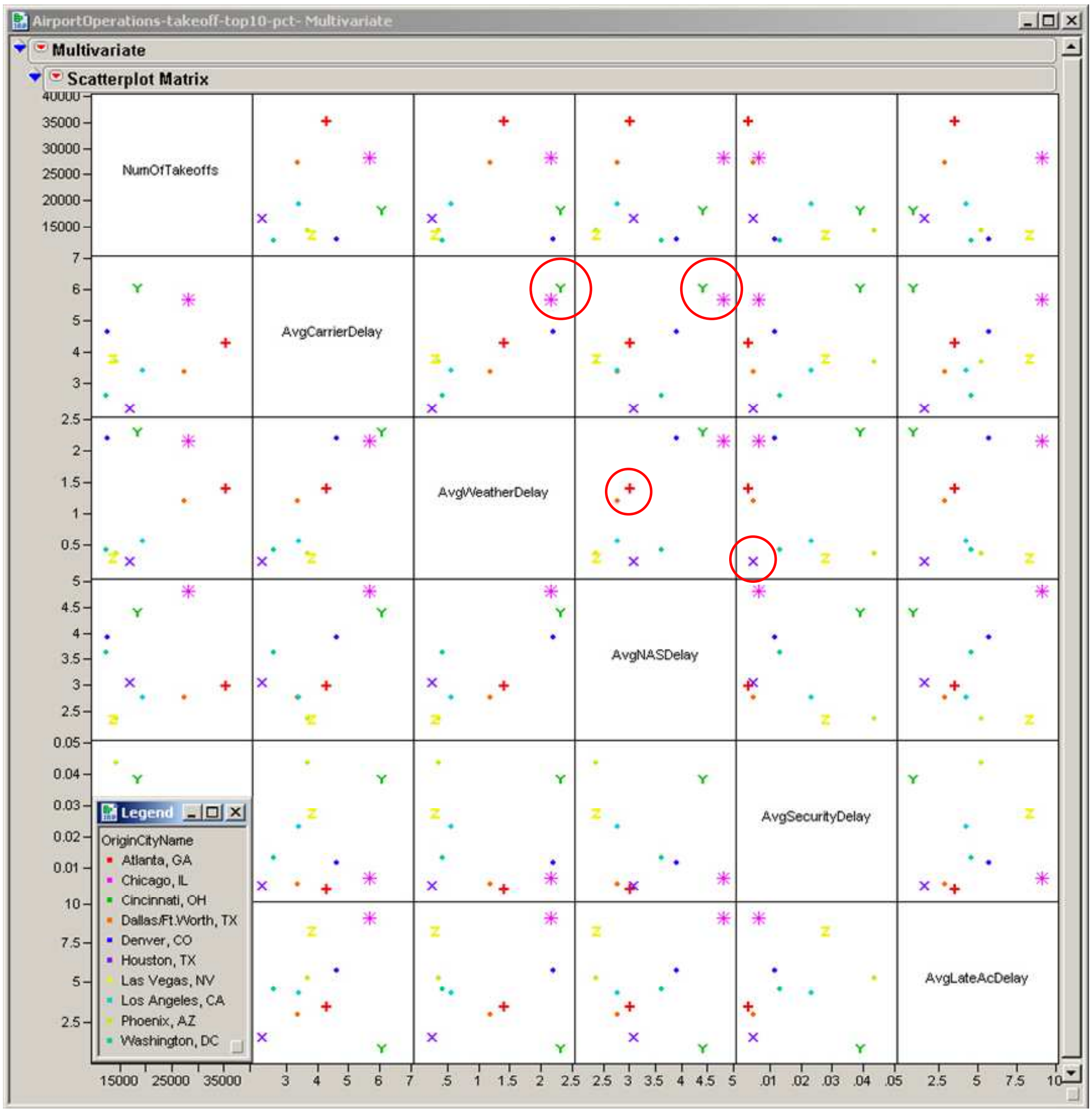



Figure 5.2.19 Delay Cause Analysis – Top 10 Big Domestic Airports

As we can see the Chicago O'Hare International Airport (ORD) and the Cincinnati/Northern Kentucky International Airport (CVG), as the point "*" and "y", are the worst performers in the top 10 airports, since they are listed at top in most of the delay categories. The Houston Intercontinental Airport (IAH), as the point “x”, operated very well in the case of most delay categories. The busiest, Atlanta Airport (ATL), as the point “+”, is among the average in most of delay causes, and shortest in security delay.

On the other hand, we can take a look at the on-time performance of the top 10 airports. We generated some features as system metrics to evaluate on time performance. PctOnTime is the percentage of on-time flights over all flights, where the term “on time” is defined as the flight departing within 30 minutes of its scheduled time. PctXXXDelay is the percentage of the delay time in a category over the total delay time, where XXX is a delay category, such as Carrier and NAS.

Examining the Figure 5.2.20, we confirm that the Houston Airport (IAH), as the point “x”, has the best on time performance, which has 93% of flights on time. The Chicago Airport (ORD) at the point “*” is the worst performer in terms of keeping flights on time (78%), and the Las Vegas McCarran International Airport (LAS) at the point “z” is the second worst with only 83% of flights on time. As for the cause of the delay, Figure 5.2.20 shows that late aircraft delay, carrier delay, and NAS delay are three causes of the delay, and each of them takes around 30% of the delay time. Weather delay takes the rest of delay time, and security delay can be neglected.

In this subchapter, we implemented the hierarchical graphic analysis approach proposed in an earlier chapter, and investigated the national air transportation system operations data to evaluate the on-time performance of the system, carriers and airports. Although we could drill further in the hierarchy into the details of each flight to find out more information, we choose not to do so in this sample due to time constraints. In addition, with only one month of data, the information could be biased, and data in a longer period could be used to reduce the possibility of prejudice.

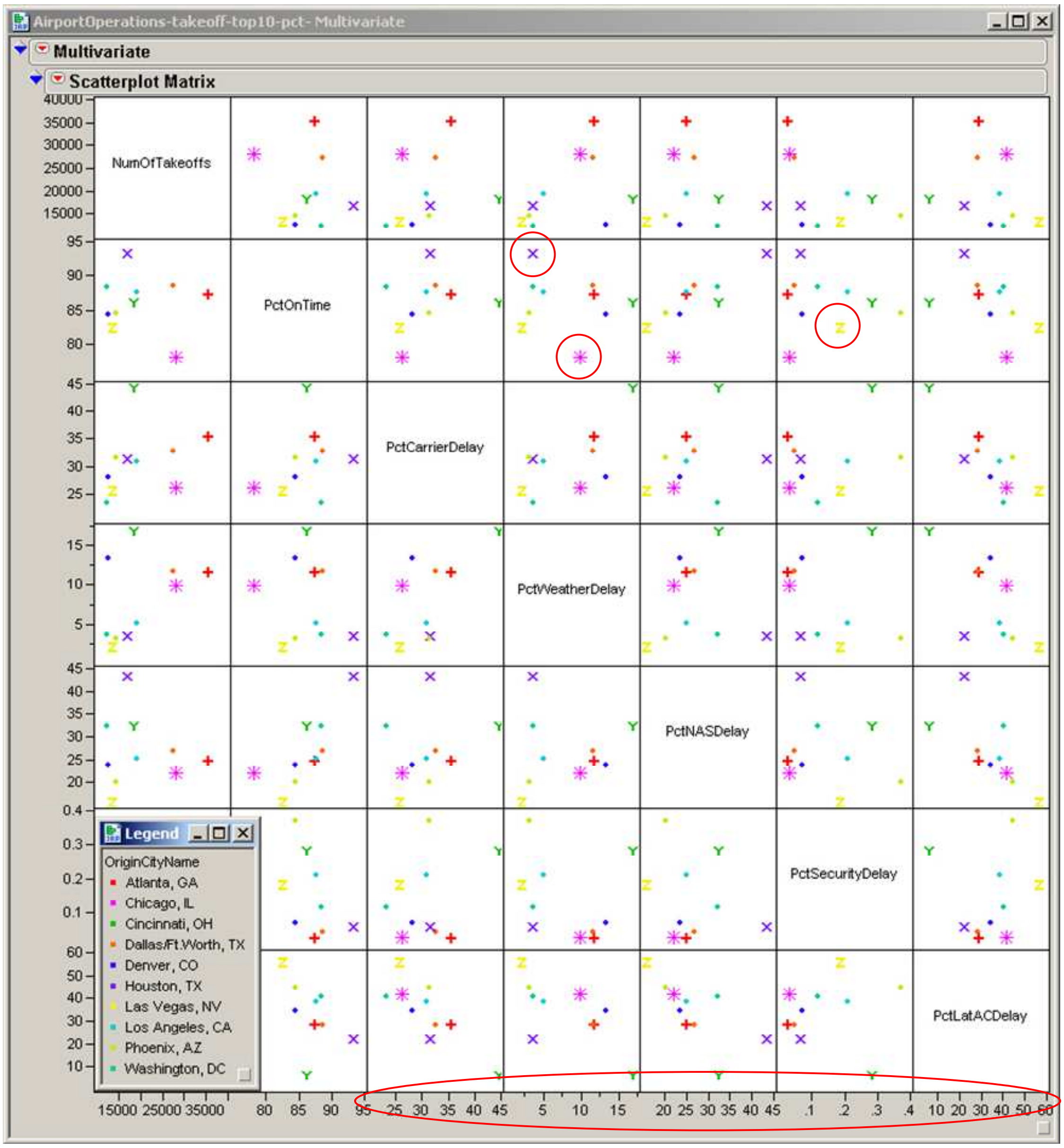



Figure 5.2.20 On-Time Operation Analysis – Top 10 Big Domestic Airports

## 5.3 Knowledge Management in DSpace

A prototype of the knowledge management environment for aircraft life cycle design decision support is created with the help of DSpace technology as depicted in Figure 5.3.1. Three top level components are created, which are aircraft design, aircraft operations, and aircraft maintenance. Some sample knowledge is entered into each of the components so that users can browse as displayed in Figure 5.3.2 and search as shown in Figure 5.3.3 for knowledge. The detailed descriptive metadata related with the knowledge will be displayed as illustrated in Figure 5.3.4, once click on a item in the search result page. The contained media files can be selected and downloaded to a local computer for in depth review.


Figure 5.3.1 Knowledge Management Structure

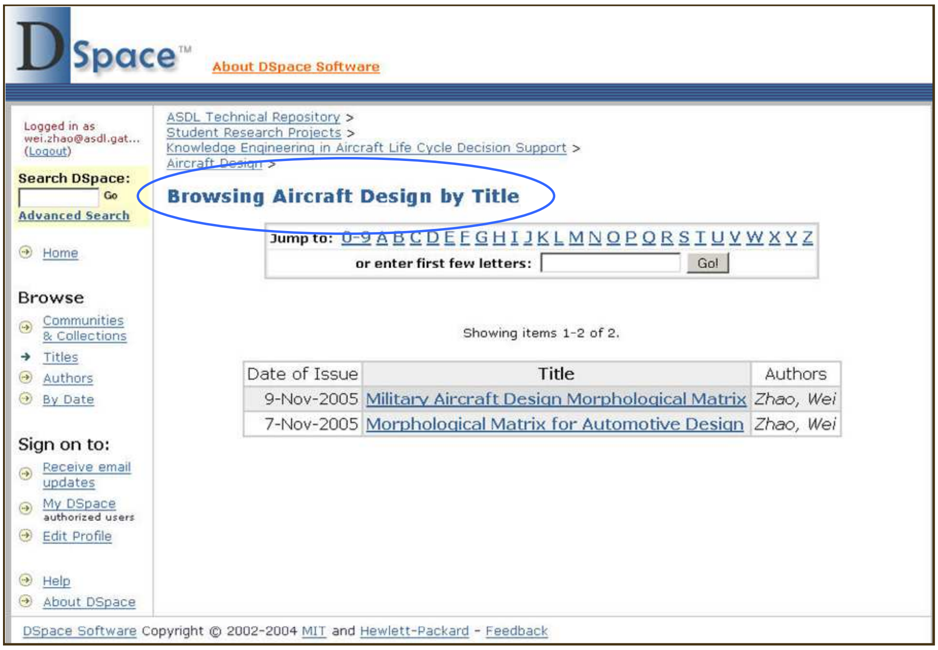



Figure 5.3.2 Browsing a Component by Title

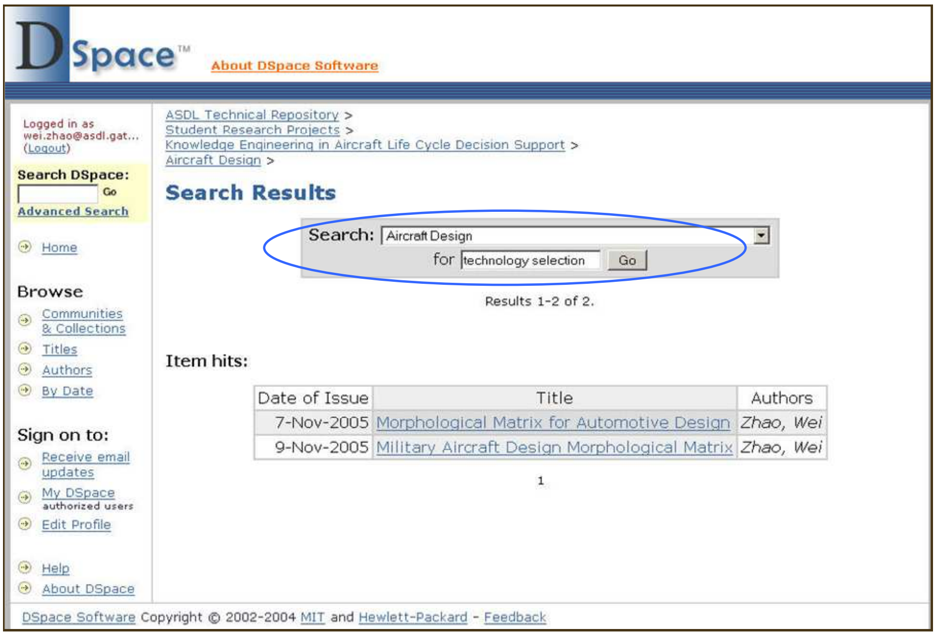



Figure 5.3.3 Search Knowledge by Keywords


Figure 5.3.4 Detailed Descriptive Metadata of the Knowledge

## 5.4 Knowledge Integration with Data Configuration Control

As discussed in section 4.5, a data configuration control process is proposed to maintain the integrity of the knowledge embedded in surrogate models. To demonstrate the process, a sample environment is created based on a landing gear structure analysis project, which is to model the damping coefficients and corresponding frequencies to prevent the landing gear's shimmy phenomena.

The data configuration control environment consists of two files. One is an Excel file with multiple tabs to store environment information, such as the model definition, variables and responses, information about the simulation codes, and the coefficients of the surrogate model; the other one is a JMP journal file, which keeps a record of the process of the surrogate model generation, such as model type, model parameter setting, model accuracy and model verification. This environment takes user inputs, and is not an automatic program which does everything for the user. However, it provides a standard template to guide user to record information so that important configurations are not lost in the modeling process, and it is flexible for the user to customize and extend the capacity according to needs.

The first tab in the Excel file is the simulation environment definition, which includes model definition and the related information for the analysis tools (see Figure 5.4.1). Some fields (underlined and in blue) have hyperlinks associated with them. When it is clicked, additional information can be brought to user, such as language references, the company's webpage, user manual, or a sample file.

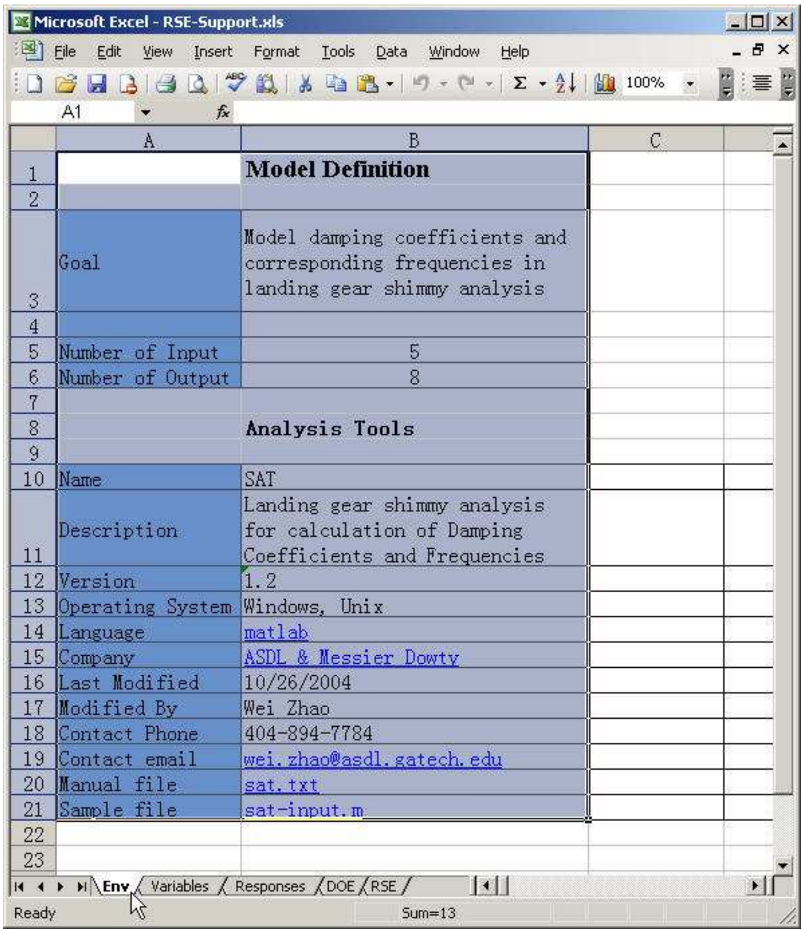



Figure 5.4.1 Simulation Environment Definition

The second tab (Figure 5.4.2) is for input variable definition, which tracks all the input variables that the user should care about. The first column is the variable name in the surrogate model, and it will be empty if the variable will not vary between cases, which means it will not be included in the surrogate model while its value is in the user's control. The next several columns are the variable name in the code, variable description, where the variable value is assigned, the format and the unit of the variable. The next a few columns are related with DoE, such as baseline value, ranges and distribution. A variable can be an array of values. The last column is a reference field, where any related information can be added via hyperlinks.

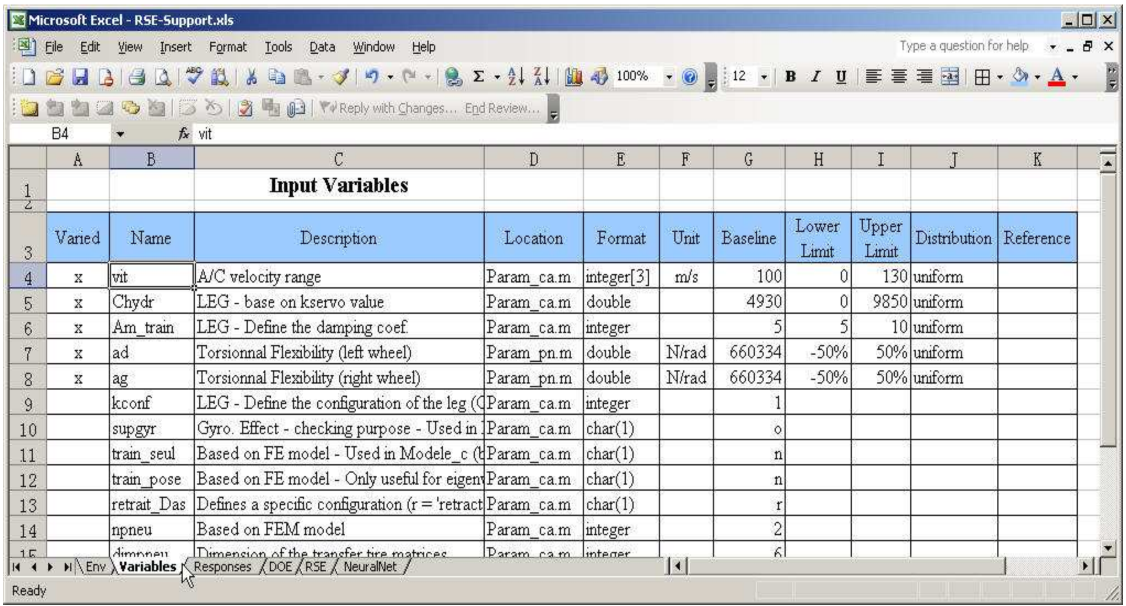
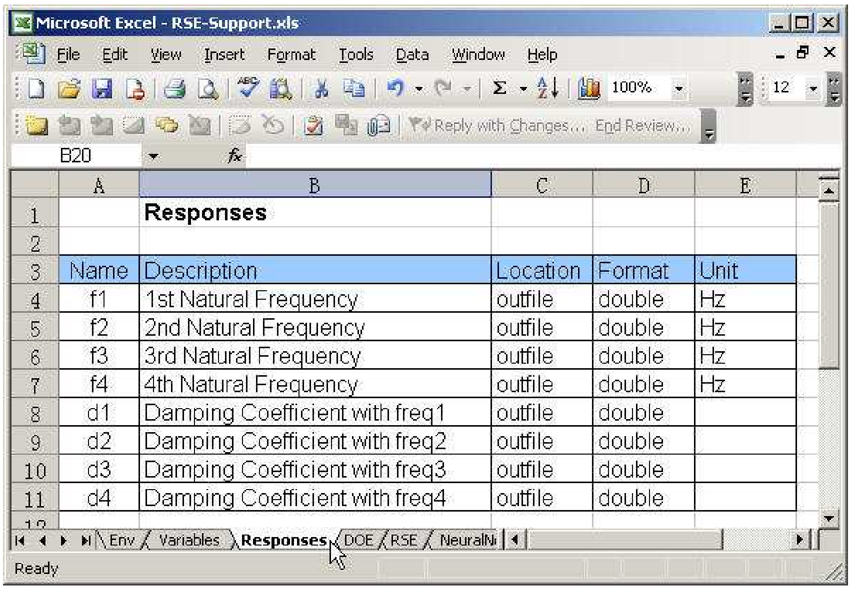



Figure 5.4.2 Input Variable Definition

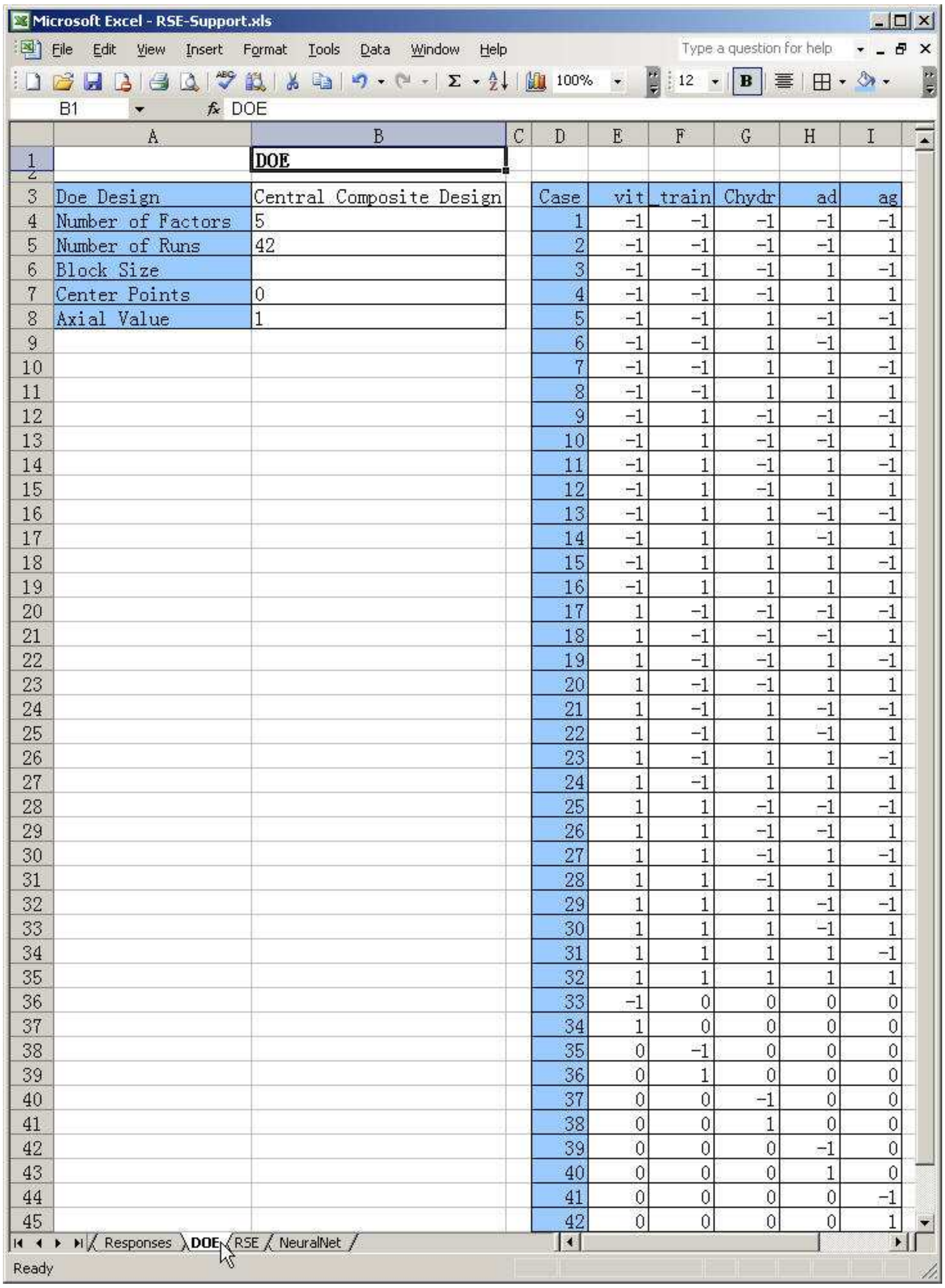
The third tab (Figure 5.4.3) is for response definition, which is similar to input variable definition and lists the information about responses we are trying to get from the simulation, and create surrogate models. Each response will have a model associated to it.



| | | | | |
|---|---|---|---|---|
| | **Responses** | | | |
| Name | Description | Location | Format | Unit |
| f1 | 1st Natural Frequency | outfile | double | Hz |
| f2 | 2nd Natural Frequency | outfile | double | Hz |
| f3 | 3rd Natural Frequency | outfile | double | Hz |
| f4 | 4th Natural Frequency | outfile | double | Hz |
| d1 | Damping Coefficient with freq1 | outfile | double | |
| d2 | Damping Coefficient with freq2 | outfile | double | |
| d3 | Damping Coefficient with freq3 | outfile | double | |
| d4 | Damping Coefficient with freq4 | outfile | double | |

Figure 5.4.3 Response Definition

The next tab (Figure 5.4.4) is for Design of Experiment (DoE) settings. It maintains the design type, design parameters, and the actual DoE table.

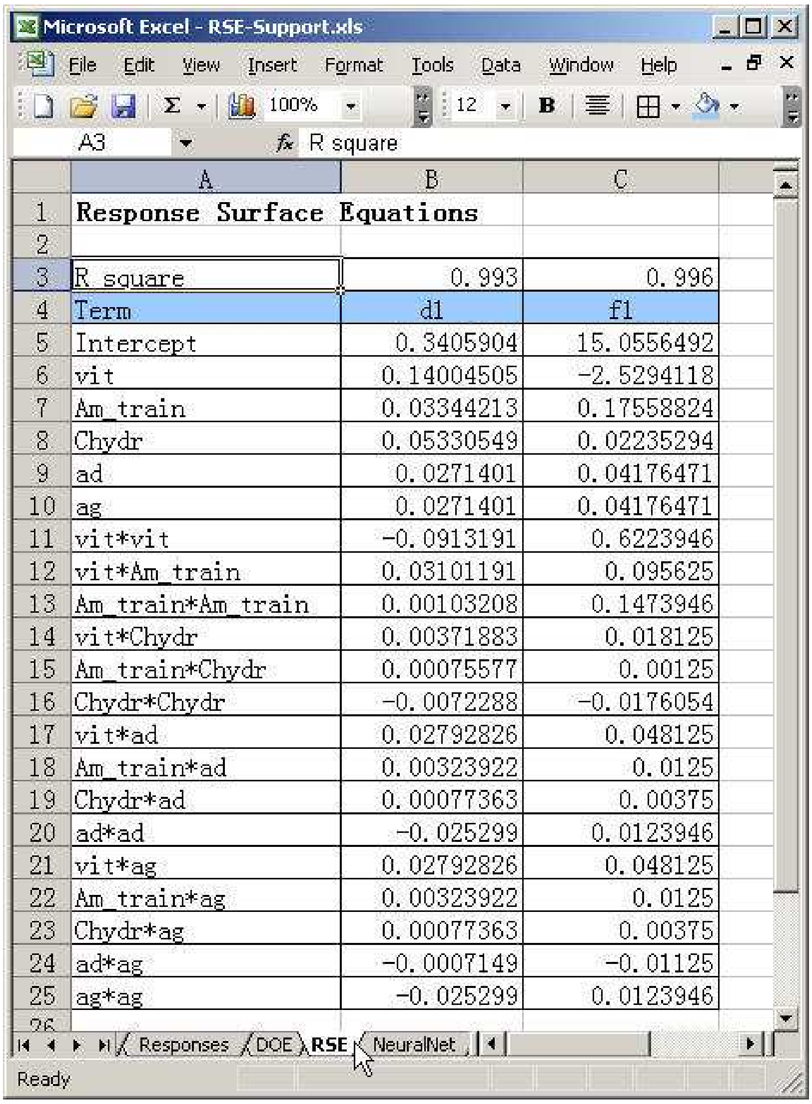



| | |
|---|---|
| | DOE |
| Doe Design | Central Composite Design |
| Number of Factors | 5 |
| Number of Runs | 42 |
| Block Size | |
| Center Points | 0 |
| Axial Value | 1 |

| Case | vit | _train | Chydr | ad | ag |
|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | -1 | -1 |
| 2 | -1 | -1 | -1 | -1 | 1 |
| 3 | -1 | -1 | -1 | 1 | -1 |
| 4 | -1 | -1 | -1 | 1 | 1 |
| 5 | -1 | -1 | 1 | -1 | -1 |
| 6 | -1 | -1 | 1 | -1 | 1 |
| 7 | -1 | -1 | 1 | 1 | -1 |
| 8 | -1 | -1 | 1 | 1 | 1 |
| 9 | -1 | 1 | -1 | -1 | -1 |
| 10 | -1 | 1 | -1 | -1 | 1 |
| 11 | -1 | 1 | -1 | 1 | -1 |
| 12 | -1 | 1 | -1 | 1 | 1 |
| 13 | -1 | 1 | 1 | -1 | -1 |
| 14 | -1 | 1 | 1 | -1 | 1 |
| 15 | -1 | 1 | 1 | 1 | -1 |
| 16 | -1 | 1 | 1 | 1 | 1 |
| 17 | 1 | -1 | -1 | -1 | -1 |
| 18 | 1 | -1 | -1 | -1 | 1 |
| 19 | 1 | -1 | -1 | 1 | -1 |
| 20 | 1 | -1 | -1 | 1 | 1 |
| 21 | 1 | -1 | 1 | -1 | -1 |
| 22 | 1 | -1 | 1 | -1 | 1 |
| 23 | 1 | -1 | 1 | 1 | -1 |
| 24 | 1 | -1 | 1 | 1 | 1 |
| 25 | 1 | 1 | -1 | -1 | -1 |
| 26 | 1 | 1 | -1 | -1 | 1 |
| 27 | 1 | 1 | -1 | 1 | -1 |
| 28 | 1 | 1 | -1 | 1 | 1 |
| 29 | 1 | 1 | 1 | -1 | -1 |
| 30 | 1 | 1 | 1 | -1 | 1 |
| 31 | 1 | 1 | 1 | 1 | -1 |
| 32 | 1 | 1 | 1 | 1 | 1 |
| 33 | -1 | 0 | 0 | 0 | 0 |
| 34 | 1 | 0 | 0 | 0 | 0 |
| 35 | 0 | -1 | 0 | 0 | 0 |
| 36 | 0 | 1 | 0 | 0 | 0 |
| 37 | 0 | 0 | -1 | 0 | 0 |
| 38 | 0 | 0 | 1 | 0 | 0 |
| 39 | 0 | 0 | 0 | -1 | 0 |
| 40 | 0 | 0 | 0 | 1 | 0 |
| 41 | 0 | 0 | 0 | 0 | -1 |
| 42 | 0 | 0 | 0 | 0 | 1 |

Figure 5.4.4 Design of Experiment Settings

After the model fitting is finished, we can store the surrogate models in the RSE tab of the Excel template for portability, since it can be used on any computer that has Microsoft Office. It can also be converted into a text file, then it can be used anywhere without the operating system restriction. Figure 5.4.5 shows the RSE tab with the model coefficients in the response surface equations for two of the responses, and the fitting R squares are also listed.



Microsoft Excel - RSE-Support.xls

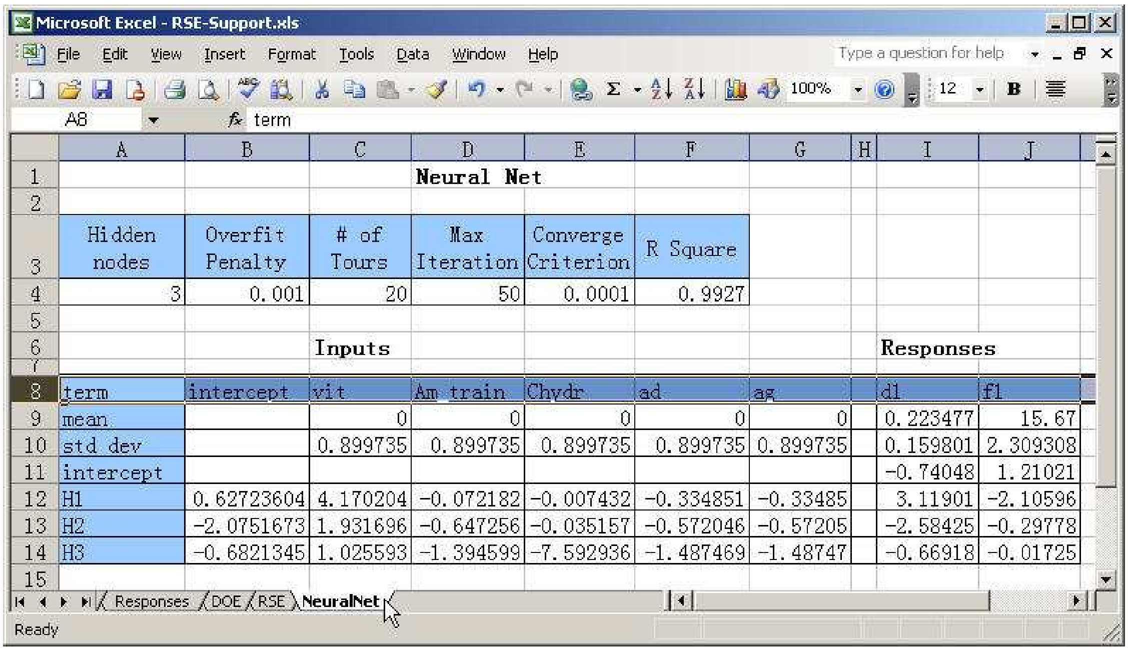
| | A | B | C |
|---|---|---|---|
| 1 | Response Surface Equations | | |
| 2 | | | |
| 3 | R square | 0.993 | 0.996 |
| 4 | Term | d1 | f1 |
| 5 | Intercept | 0.3405904 | 15.0556492 |
| 6 | vit | 0.14004505 | -2.5294118 |
| 7 | Am_train | 0.03344213 | 0.17558824 |
| 8 | Chydr | 0.05330549 | 0.02235294 |
| 9 | ad | 0.0271401 | 0.04176471 |
| 10 | ag | 0.0271401 | 0.04176471 |
| 11 | vit*vit | -0.0913191 | 0.6223946 |
| 12 | vit*Am_train | 0.03101191 | 0.095625 |
| 13 | Am_train*Am_train | 0.00103208 | 0.1473946 |
| 14 | vit*Chydr | 0.00371883 | 0.018125 |
| 15 | Am_train*Chydr | 0.00075577 | 0.00125 |
| 16 | Chydr*Chydr | -0.0072288 | -0.0176054 |
| 17 | vit*ad | 0.02792826 | 0.048125 |
| 18 | Am_train*ad | 0.00323922 | 0.0125 |
| 19 | Chydr*ad | 0.00077363 | 0.00375 |
| 20 | ad*ad | -0.025299 | 0.0123946 |
| 21 | vit*ag | 0.02792826 | 0.048125 |
| 22 | Am_train*ag | 0.00323922 | 0.0125 |
| 23 | Chydr*ag | 0.00077363 | 0.00375 |
| 24 | ad*ag | -0.0007149 | -0.01125 |
| 25 | ag*ag | -0.025299 | 0.0123946 |

Responses / DOE / RSE / NeuralNet

Ready

Figure 5.4.5 Response Surface Model

In the case of the Neural Network model, we use the NeuralNet tab to store the model formula, Figure 5.4.6. The upper part of the tab summarizes the model fit settings, such as the number of hidden nodes which is the most important number for fitting quality; overfit penalty which helps to prevent the model from overfitting; number of tours which is the number of individual model fits at different random starting values to prevent local minima and increase the likelihood of finding global minima; max iteration which sets the maximum number of iteration before each tour reporting non-convergence; Converge criterion is the relative change in the objective function that an iteration must meet to be treated as converged.




| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | Neural Net | | | | | | |
| 2 | | | | | | | | | | |
| 3 | Hidden nodes | Overfit Penalty | # of Tours | Max Iteration | Converge Criterion | R Square | | | | |
| 4 | 3 | 0.001 | 20 | 50 | 0.0001 | 0.9927 | | | | |
| 5 | | | | | | | | | | |
| 6 | | | Inputs | | | | | | Responses | |
| 7 | | | | | | | | | | |
| 8 | term | intercept | vit | Am train | Chydr | ad | ag | | d1 | f1 |
| 9 | mean | | 0 | 0 | 0 | 0 | 0 | | 0.223477 | 15.67 |
| 10 | std dev | | 0.899735 | 0.899735 | 0.899735 | 0.899735 | 0.899735 | | 0.159801 | 2.309308 |
| 11 | intercept | | | | | | | | -0.74048 | 1.21021 |
| 12 | H1 | 0.62723604 | 4.170204 | -0.072182 | -0.007432 | -0.334851 | -0.33485 | | 3.11901 | -2.10596 |
| 13 | H2 | -2.0751673 | 1.931696 | -0.647256 | -0.035157 | -0.572046 | -0.57205 | | -2.58425 | -0.29778 |
| 14 | H3 | -0.6821345 | 1.025593 | -1.394599 | -7.592936 | -1.487469 | -1.48747 | | -0.66918 | -0.01725 |
| 15 | | | | | | | | | | |

Figure 5.4.6 Neural Network Model

The lower part of the tab stores the coefficients of the neural network model. In JMP, it uses the following equations to build the neural net work model.

$$R_i = [a_{r_i} + \sum_{j=1}^{N} (c_{H_j} \times H_j)] \times \sigma_{r_i} + \mu_{r_i}$$

$$H_j = Squish\{ b_{x_i} + \sum_{k=1}^{M} [c_{x_k} \times \frac{(x_k - \mu_{x_k})}{\sigma_{x_k}}]\}$$

Where

$R_i$ is the output model for response variable $r_i$,

$H_j$ is the hidden functions built from input variable $x_k$,

σ and μ are the standard deviation and mean of a variable,

a, b, c are model coefficients we are storing in the Excel file,

M and N are the number of input variables and the number of hidden nodes

Squish() is a S-shaped activation function, which is the logistic function in JMP

$$Squish(x) = \frac{1}{1+e^{-x}}$$

For the sample case in Figure 5.4.6, we have

d1=[-0.74048+(3.11901*H1-2.58425*H2-0.66918*H3)]*0.159801+0.223477

H1=Squish{0.62723604+[4.170204*(vit-0)/0.899735-0.072182*(Am_train-0)/0.899735-0.007432*(Chydr-0)/0.899735-0.334851*(ad-0)/0.899735-0.33485*(ag-0)/0.899735]}

$$H1 = Squish\{0.62723604 + ( \begin{array}{r} 4.170204 * \frac{vit - 0}{0.899735} \\ - 0.072182 * \frac{Am\_train - 0}{0.899735} \\ - 0.007432 * \frac{Chydr - 0}{0.899735} \\ - 0.334851 * \frac{ad - 0}{0.899735} \\ - 0.33485 * \frac{ag - 0}{0.899735} \end{array} )\}$$

The actual surrogate model creation process can be recorded in JMP as journals. When we have a step finished in the process, we highlight the window we want to record and select Edit → Journal from the JMP tool menu, then the information in current window will be copied into a journal window, which can be edited and saved for future reference.

Figure 5.4.7 illustrates the DoE table and response values for each of the cases. A hyperlink to the previous discussed Excel template file is also created under the section of model description.

**DoE Table**

| Pattern | vit | Am_train | Chydr | ad | ag | d1 | f1 |
|---|---|---|---|---|---|---|---|
| ----- | -1 | -1 | -1 | -1 | -1 | -0.000007088265 | 18.39 |
| ----+ | -1 | -1 | -1 | -1 | 1 | 0.0001168691 | 18.29 |
| ---+- | -1 | -1 | -1 | 1 | -1 | 0.0001168691 | 18.29 |
| ---++ | -1 | -1 | -1 | 1 | 1 | 0.0001310756 | 18.11 |
| --+-- | -1 | -1 | 1 | -1 | -1 | 0.1002101 | 18.39 |
| --+-+ | -1 | -1 | 1 | -1 | 1 | 0.09786749 | 18.31 |
| --++- | -1 | -1 | 1 | 1 | -1 | 0.09786749 | 18.31 |
| --+++ | -1 | -1 | 1 | 1 | 1 | 0.09567305 | 18.13 |
| -+--- | -1 | 1 | -1 | -1 | -1 | 0.00008129828 | 18.43 |
| -+--+ | -1 | 1 | -1 | -1 | 1 | 0.00006341739 | 18.43 |
| -+-+- | -1 | 1 | -1 | 1 | -1 | 0.00006341739 | 18.43 |
| -+-++ | -1 | 1 | -1 | 1 | 1 | 0.00004520929 | 18.43 |
| -++-- | -1 | 1 | 1 | -1 | -1 | 0.09999791 | 18.43 |
| -++-+ | -1 | 1 | 1 | -1 | 1 | 0.09997847 | 18.43 |
| -+++- | -1 | 1 | 1 | 1 | -1 | 0.09997847 | 18.43 |
| -++++ | -1 | 1 | 1 | 1 | 1 | 0.09995829 | 18.43 |
| +---- | 1 | -1 | -1 | -1 | -1 | 0.1213482 | 12.88 |

Figure 5.4.7 DoE and Responses in JMP journal

Figure 5.4.8 records the model specification for a response surface model, which includes the responses selected for model generation, a list of model effects, and other model settings, such as personality and emphasis.

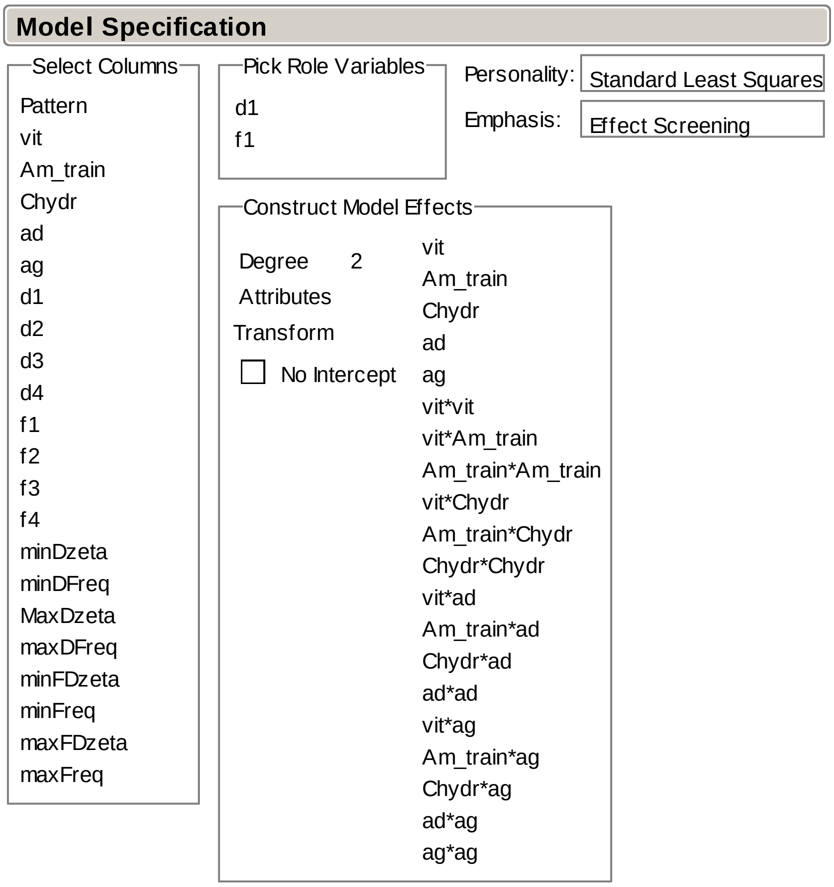
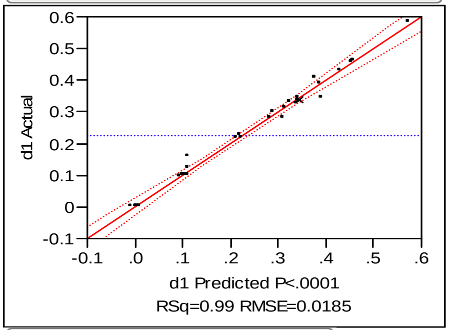



Figure 5.4.8 Model Specification in JMP Journal

For a response surface model, Figure 5.4.9 displays the model summary for a sample response, d1. Model coefficients are listed in the Parameter Estimates section with some statistics, such as standard error, t ratio, and so on. Fitting summary and some row diagnostic plots, such as Actual vs. predicted plot, are also stored in this journal file.

**Response d1**

**Actual by Predicted Plot**




**Summary of Fit**

| | |
|---|---|
| RSquare | 0.992995 |
| RSquare Adj | 0.986626 |
| Root Mean Square Error | 0.01848 |
| Mean of Response | 0.223477 |
| Observations (or Sum Wgts) | 43 |

**Analysis of Variance**

| Source | DF | Sum of Squares | Mean Square | F Ratio |
|---|---|---|---|---|
| Model | 20 | 1.0650180 | 0.053251 | 155.9227 |
| Error | 22 | 0.0075135 | 0.000342 | Prob > F |
| C. Total | 42 | 1.0725315 | | <.0001* |

**Parameter Estimates**

| Term | Estimate | Std Error | t Ratio | Prob>\|t\| |
|---|---|---|---|---|
| Intercept | 0.3405904 | 0.00683 | 49.87 | <.0001* |
| vit | 0.1400451 | 0.003169 | 44.19 | <.0001* |
| Am_train | 0.0334421 | 0.003169 | 10.55 | <.0001* |
| Chydr | 0.0533055 | 0.003169 | 16.82 | <.0001* |
| ad | 0.0271401 | 0.003169 | 8.56 | <.0001* |
| ag | 0.0271401 | 0.003169 | 8.56 | <.0001* |
| vit*vit | -0.091319 | 0.011793 | -7.74 | <.0001* |
| vit*Am_train | 0.0310119 | 0.003267 | 9.49 | <.0001* |
| Am_train*Am_train | 0.0010321 | 0.011793 | 0.09 | 0.9311 |
| vit*Chydr | 0.0037188 | 0.003267 | 1.14 | 0.2672 |
| Am_train*Chydr | 0.0007558 | 0.003267 | 0.23 | 0.8192 |
| Chydr*Chydr | -0.007229 | 0.011793 | -0.61 | 0.5462 |
| vit*ad | 0.0279283 | 0.003267 | 8.55 | <.0001* |
| Am_train*ad | 0.0032392 | 0.003267 | 0.99 | 0.3322 |
| Chydr*ad | 0.0007736 | 0.003267 | 0.24 | 0.8150 |
| ad*ad | -0.025299 | 0.011793 | -2.15 | 0.0432* |
| vit*ag | 0.0279283 | 0.003267 | 8.55 | <.0001* |
| Am_train*ag | 0.0032392 | 0.003267 | 0.99 | 0.3322 |
| Chydr*ag | 0.0007736 | 0.003267 | 0.24 | 0.8150 |
| ad*ag | -0.000715 | 0.003267 | -0.22 | 0.8288 |
| ag*ag | -0.025299 | 0.011793 | -2.15 | 0.0432* |

Figure 5.4.9 Sample RSE Model Summary

In addition, the factor profilers, such as the prediction profiler, contour profiler, and surface profiler, can be included as snapshots for visual display, Figure 5.4.10.

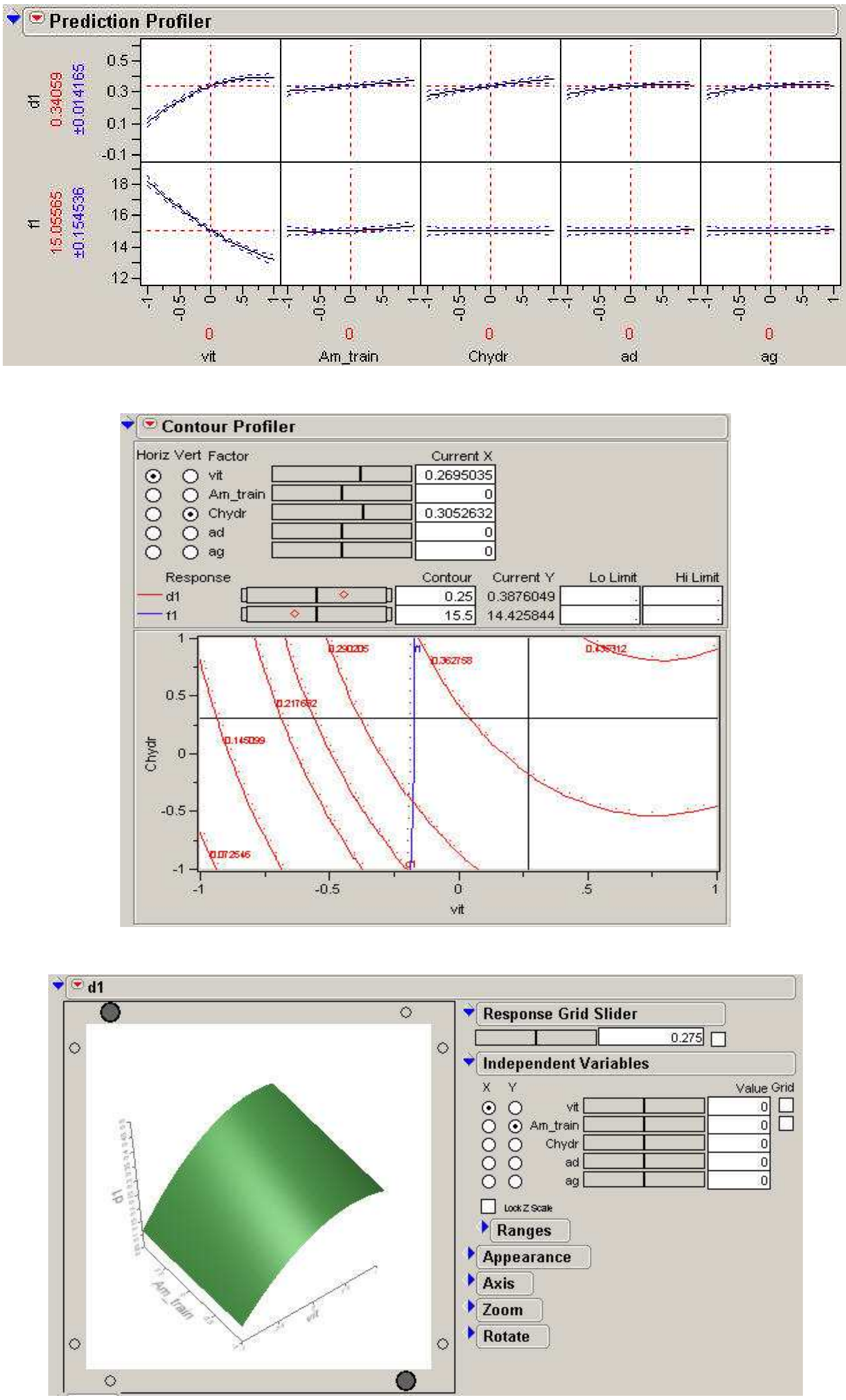



Figure 5.4.10 Sample Factor Profilers

For a Neural Network model, Figure 5.4.11 displays the sample model fitting summary for the responses of d1 and f1. Model settings are stored in the Control Panel section, and coefficients are stored in Parameter Estimates section. Row diagnostic plots and factor profilers can be included as well, although they can not be displayed in the figure.

**Neural Net**

**Control Panel**

| | Specify |
|---|---|
| Hidden Nodes | 3 |
| Overfit Penalty | 0.001 |
| Number of Tours | 20 |
| Max Iterations | 50 |
| Converge Criterion | 0.00001 |

No Cross Validation

**Fit History**

| Nodes | Penalty | RSquare |
|---|---|---|
| 3 | 0.001 | 0.99267 |

**Current Fit Results**

| | Objective |
|---|---|
| SSE | 0.6155345588 |
| Penalty | 0.1140442469 |
| Total | 0.7295788056 |
| N | 43 |
| Nparm | 26 |

| | |
|---|---|
| 8 | Converged At Best |
| 12 | Converged Worse Than Best |
| 0 | Stuck on Flat |
| 0 | Failed to Improve |
| 0 | Reached Max Iter |

| Y | SSE | RMSE | SSE Scaled | RSME Scaled | RSquare |
|---|---|---|---|---|---|
| d1 | 0.010547499 | 0.01644532 | 0.4130367967 | 0.10291106 | 0.9902 |
| f1 | 1.0799012795 | 0.16640245 | 0.202497762 | 0.07205727 | 0.9952 |

**Parameter Estimates**

| Parameter | Estimate |
|---|---|
| H1:Intercept | 0.6272360425 |
| H2:Intercept | -2.075167331 |
| H3:Intercept | -0.682134522 |
| H1:vit | 4.1702037326 |
| H1:Am_train | -0.072182422 |
| H1:Chydr | -0.007432044 |
| H1:ad | -0.334850474 |
| H1:ag | -0.334850474 |
| H2:vit | 1.9316956837 |
| H2:Am_train | -0.64725578 |
| H2:Chydr | -0.035156574 |
| H2:ad | -0.572046448 |
| H2:ag | -0.572046448 |
| H3:vit | 1.0255934804 |
| H3:Am_train | -1.394599177 |
| H3:Chydr | -7.592935857 |
| H3:ad | -1.487469109 |
| H3:ag | -1.487469109 |
| d1:Intercept | -0.74047658 |
| f1:Intercept | 1.2102098768 |
| d1:H1 | 3.1190098668 |
| d1:H2 | -2.584254329 |
| d1:H3 | -0.669176877 |
| f1:H1 | -2.105959769 |
| f1:H2 | -0.297783105 |
| f1:H3 | -0.017247425 |

Random Seed
1424183373

Figure 5.4.11 Neural Network Model Summary

In this sub chapter, we demonstrated the implementation of the data configuration control process in a sample environment created for landing gear shimmy analysis. As we have seen, complete information on simulation environment, model setting, and the surrogate models is captured in an organized manner. With this information as reference, we can recreate or update model when necessary, which includes reusing the case results already generated, keep consistent with previous modeling by using the same model settings, and compare the model results and accuracy with previous models. The reusability of the model is improved significantly.

## 5.5 Knowledge Maps of the Aircraft Life cycle

Knowledge maps of the aircraft life cycle can be presented in a similar structure as the human's mind map, see Figure 5.5.1. The map is developed in OpenMind. Centered with an aircraft as the subject, we can attach all the phases of the aircraft life cycle to it, such as design, production, operations, support, and marketing. The information can be in many formats, such as text descriptions, pictures, audios, videos, hypertext, and so on.

As a branch of the mind map, each aircraft phase can have its own branches and sub-branches with detailed information. And we can expend the branches to get in-depth knowledge, or collapse branches for brief top views. For example, the layout of the Boeing 777 is attached to the Design→Conceptual Design→ Design Tasks→Layout, and one can click the thumbnail to see the larger picture. A weight breakdown chart of different materials on Boeing 777 is attached to Design→Conceptual Design→ Design Tasks→ Performance→Weight.

The knowledge map can be used as a central access point for all the aircraft life cycle related knowledge. From this map, a user can go deep into a branch to view more detailed knowledge when making the design decision. For example, if he is interested in the design technology alternatives in the conceptual design phase, he can follow the link Design →Conceptual Design→Design Space Exploration→Morphological Matrix to display the morphological matrix associated with this aircraft type, and view all the listed technologies for each of the functional area. If he is not sure about a technology concept, he can click on the technology, and get more detailed information and explanation.

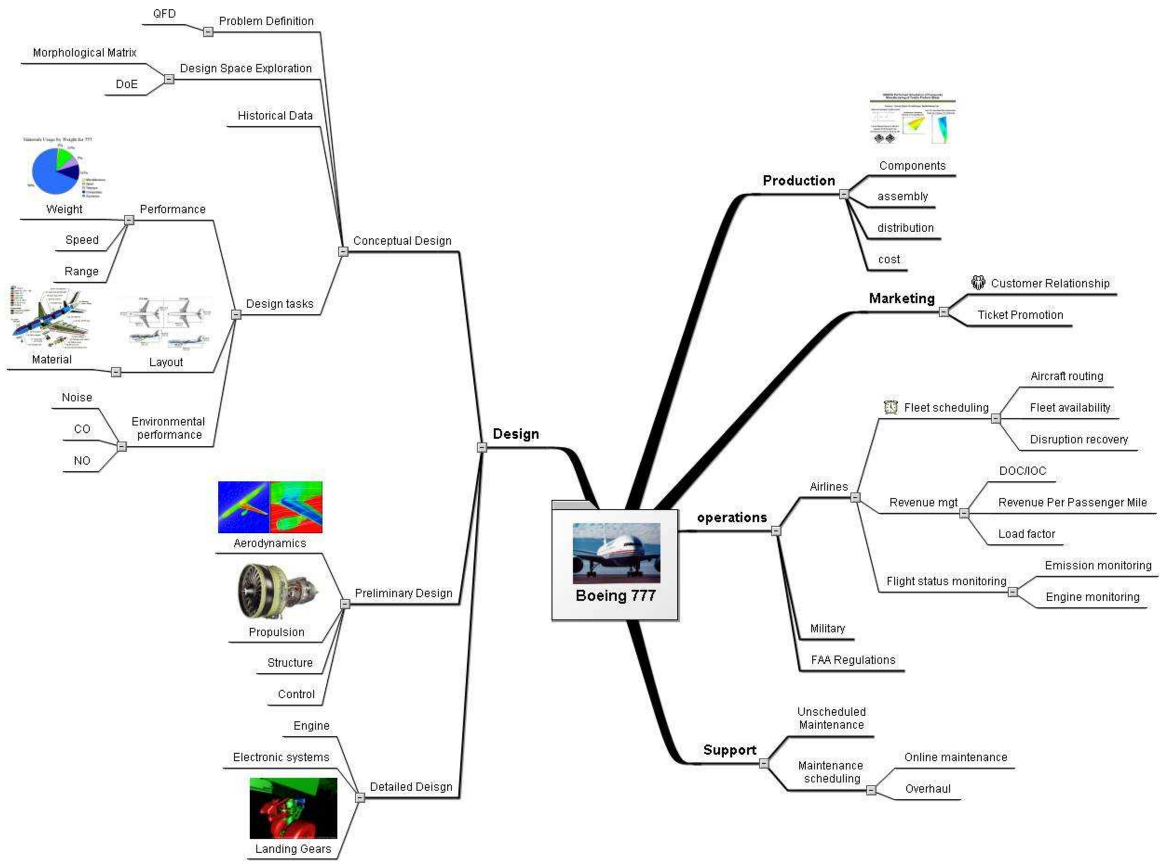



Figure 5.5.1 Knowledge Maps of the Aircraft Life cycle

## 5.6 Design Space Exploration with Knowledge-enabled Multimedia Morphological Matrix

In the aircraft life cycle, the concept generation phase is an important stage of the aircraft design process. This phase dictates the level of innovation and predetermines the aircraft cost to a significant extent. By dividing the design task into smaller tasks, these methodologies try to narrow the cognitive effort and focus on the innovative thinking; and then, to create several solutions to address each function. Once the solutions are generated, an overall solution is synthesized by identifying individual solutions for each function that are compatible. This is the core of the morphological matrix.

The morphological matrix is a methodology for organizing alternative solutions for each function of a system and combining them to generate a great number of solution variants each of which can potentially satisfy the system level design need. The morphological matrix methodology is an excellent way to record the solutions for the relevant functions and aid in the cognitive process of generating the system-level design solution. However, it will be difficult to utilize the methodology if a user is not familiar with the concepts listed in the table. To make the morphological matrix more informative and infuse knowledge into the morphological matrix so that it can be more practical to the users, we propose the knowledge-enabled multimedia morphological matrix.

With the current information technology, we can easily manage and present more information to decision maker. The technologies we used here combined the feature of the morphological matrix with multimedia and hyperlinks. With multimedia, knowledge can be presented to all human cognitive channels, such as photos, audio, video, and even interactive features. For example, if one does not know what the turbine engine is, a movie of turbine engine introduction or an animation of turbine engine at work will definitely make him/her quickly grasp the idea. With hyperlink, knowledge can be presented in an organized and easy-to-access structure. The resources are available at the user's fingertips, and can be accessed with a single mouse click.

A proof of concept example on military aircraft design is shown in Excel (Figure 5.6.1). If a user is not familiar with the concept Swing Wing, he or she can click on the cell, named "Swing Wing". A video will play in a popular media player to tell a story of the swing wing technology, and detailed explain the advantages and disadvantages of the Swing Wing with the current status of the technology.

**Functional Decomposition of Military Aircraft System**

| Characteristics | | Alternative 1 | Alternative 2 | Alternative 3 | Alternative 4 |
|---|---|---|---|---|---|
| config | Wing | Straight Wing | Swept Wing | Swing Wing | Delta Wing |
| | Fuselage | Cylindrical | Area Ruled | Oval | |
| | Pilot Visibility | Synthetic Vision | Conventional | Conventional & Nose Droop | |
| Mission | Range (nmi) | 5000 | 6000 | 6500 | |
| | Mach Number | 2 | 2.2 | 2.4 | |
| Propulsion | Type | Turbine Bypass Engine (TBE) | Variable Cycle Engine (VCE) | Fan-on-blade (Flade) | Mixed Flow Turbofan (MFTF) |
| | Materials | Conventional | High T Comp | | |
| | Combustor | Conventional | RQL | LPP | |
| | Secondary Power | Auxiliary Power Units (APU) | Fuel Cell (FC) | Hybrid | |
| | Nozzle | Conventional | Internal Flow Alternation | Mixed Ejector | Mixed Ejectot & Acoustic Linear |
| Aero | Low Speed | Conventional Flaps | Conventional Flaps & Slots | CC | |
| | High Speed | Conventional | NLFC | Active Control | HLFC |
| Struct | Materials | Aluminum | Titanium | High Temp Composite | |
| | Process | Integrally Stiffened | Spanwise Stiffened | Monocoque | Hybrid |

Figure 5.6.1 Knowledge-based Morphological Matrix for Aircraft Design

If a user does not know what is a “Nose Droop”, a picture of a aircraft with Nose Droop and some description can quickly reveal the concept, Figure 5.6.2.


Figure 5.6.2 Illustration of Nose Droop – Concorde

Some text description can be useful to clarify technology terms, such as TBE (Turbine Bypass Engine), or VCE (Variable Cycle Engine). If the mouse is hovering on the corresponding cell, a small text box will popup and display a brief explanation of the technology as displayed in Figure 5.6.3.

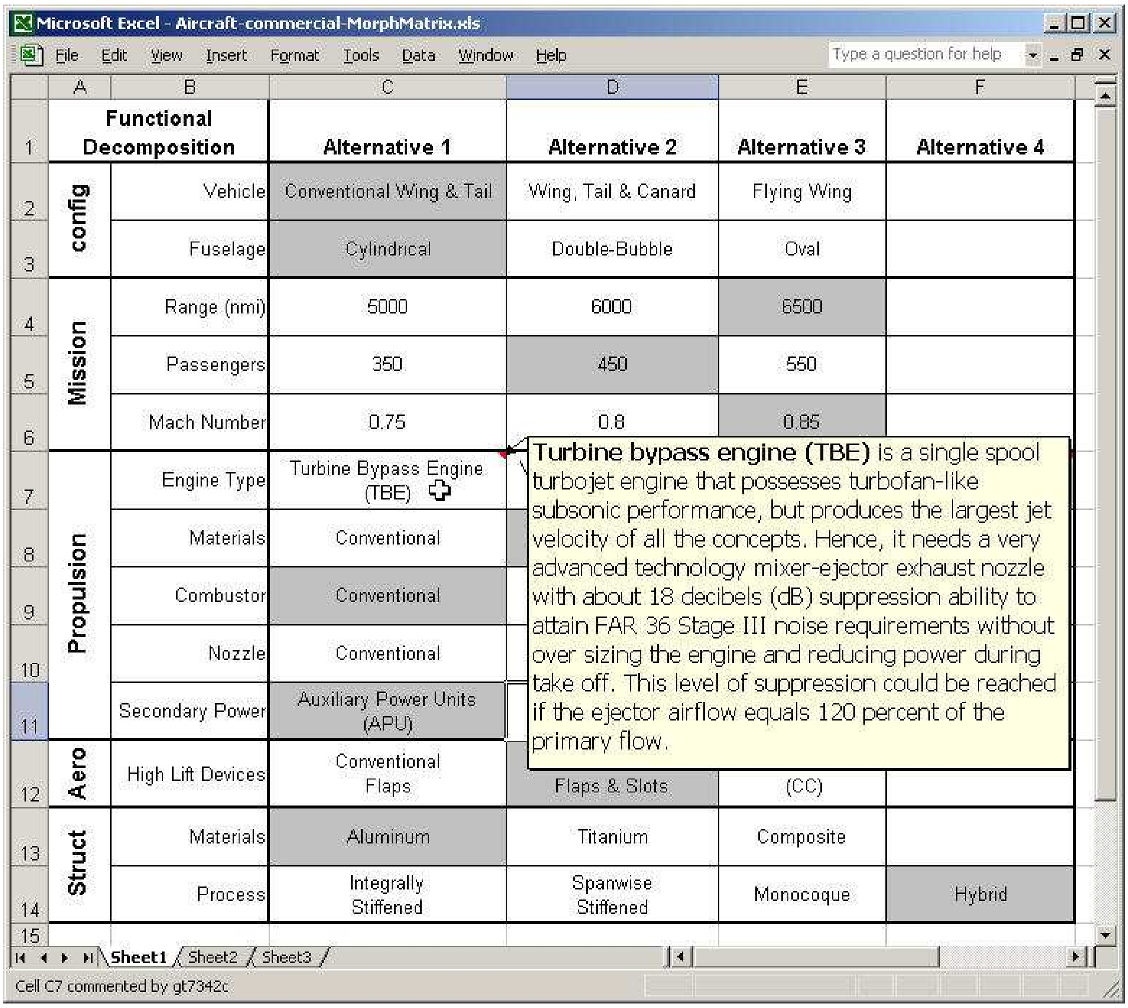



Figure 5.6.3 Popup Text Description of Terminologies

Animation is also used to demonstrate how the "fuel cell" works in Figure 5.6.4, when user clicks on the corresponding cell.

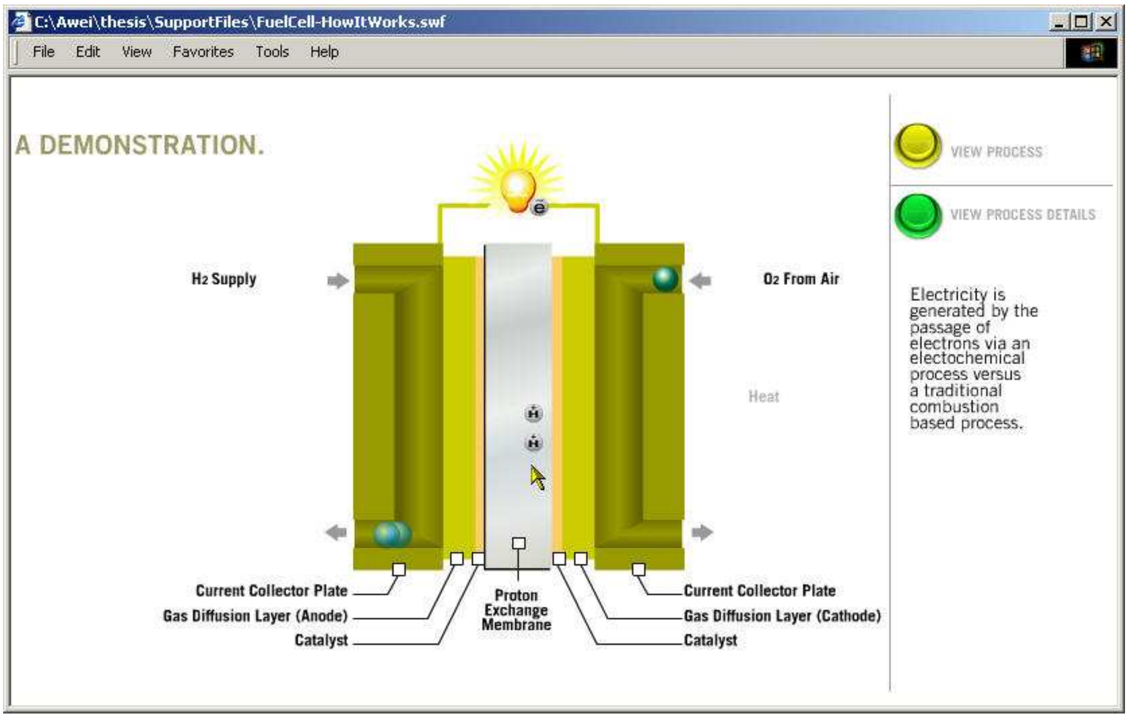



Figure 5.6.4 An Animated Demonstration on How Fuel Cell Works

The resource on the internet can also be linked to the morphological matrix to enrich the knowledge presentation. For instance, when user clicks on the RQL (Rich-Burn/Quick-Mix/Lean-Burn Combustor), a webpage will open in a web browser to display relative information of this technology, Figure 5.6.5.

The knowledge-based morphological matrix uses multimedia to support the knowledge presentation, which is not limited to the formats listed in the above sample. Any digitized information, such as an Excel spreadsheet, or even a detailed FEA model, can be linked to the matrix to enrich the knowledge and support decision making.


Figure 5.6.5 An Online Site About RQL Technology

# CHAPTER VI
# CONCLUSIONS AND RECOMMENDATIONS

To create better designs, it is critical to develop a methodology to discover, retain, organize, and present knowledge throughout all phases of the aircraft life cycle. This methodology must consider knowledge developed in the aircraft design, production, operations, and ground support and be flexible enough to capture the inherent variability of the system. The definition of this need leads to a series of research questions that this thesis attempted to resolve.

This dissertation has discussed the life cycle activities of aircraft, and examined the needs to bring knowledge back to design from the day-to-day use of the aircraft, such as operations and support. A knowledge engineering framework was proposed with multiple components, which include knowledge discovery, knowledge integration with data configuration control for surrogate models, knowledge management, and knowledge presentation. The unique process of feature-based hierarchical knowledge discovery was designed to provide greater insight into the huge amount of data existing in each of the aircraft life cycle phases. The template and process of the data configuration control is to maintain the completeness of knowledge associated with surrogate models. The knowledge management component establishes a hierarchical knowledge management structure and provides an internet enabled central place to store and organize knowledge and information. Additionally, with the proposed knowledge presentation approaches, knowledge can be presented dynamically and transferred to decision makers easily. The feasibility of the framework has been demonstrated in several phases of the aircraft life cycle activities, i.e. aircraft design, maintenance, and operations.

## 6.1 Research Questions Answered

Question 1: *How does one obtain knowledge from existing data in aircraft life cycle activities?*

With current information technology, the aerospace industry has accumulated great amount of data, and the volume is growing quickly, which causes difficulties in analyzing the existing data and extracting useful knowledge. A feature-based hierarchical knowledge discovery process is proposed to tackle this problem. We first generate a list of essential metrics from the data, called features, based on a proposed algorithm; and then utilize the features to systematically analyze the data. Data are first integrated from a lower level of detail to the higher level of abstraction in order to get a better overall understanding of the whole picture with the bottom-up analysis, and then a top-down analysis is applied to focus down to desired detail. With the proposed approach, the data in the analysis process are reduced from large scale to a smaller manageable size and focus is maintained at a proper level depending on the amount of detail needed.

Question 2: *How does one manage explicit knowledge efficiently throughout the aircraft life cycle in order to make it easily accessible?*

In the current era of the aerospace industry, information is extremely rich in content. Knowledge regarding the aircraft is developed throughout its life cycle and presents a challenge for effective management. This thesis attempted to answer Question 2 with the following steps. We first addressed the required functions of effective knowledge management and access, and then defined the hierarchy of the knowledge based on the nature of the aircraft life cycle. A framework for the knowledge management was proposed, and a sample environment was created as a proof of concept.

In the environment, information is stored in a central location with the associated meta-data, which defines the essential content associated to the main entry. Under controlled access, users can enter and search information by the pre-defined hierarchy or keyword from a remote location.

A special category of knowledge management exists for engineering design problems that require expensive simulation and optimization, which are sometimes associated with thousands or even millions of evaluations. Surrogate modeling is an effective and popular way to alleviate the computational cost by approximating the actual function with constructed algebraic expressions. The creation of a surrogate model often requires an expert team to work together. It is common that the focus is on creation of the surrogate model and the data configuration settings around the model are not maintained, or maintained separately by individuals for their own purposes. This traditional practice does not maintain complete information of the surrogate model, since it only keeps the model itself, which results in the difficulties in recreating the model when updates are necessary. A large amount of effort has to be applied again and again to obtain slightly different models, even when there are minor changes in the data. The inefficiency in this process consists of two aspects: one is the need to rerun the simulation cases to build the dataset for the new model generation although most of the cases are the same as the previous model; the other is the model generation process itself, where great effort and expert involvement are needed to create the right parametric configuration of the model to achieve specified accuracy. This thesis examined the general modeling process, and established a common data structure that includes the information about the simulation environment and model configuration. Then we suggested a process in parallel to the surrogate modeling, and provided a standard template to guide users to record the important configuration data while the model creation is in progress. With the proposed process, the integrity of the surrogate model can be well maintained in an organized manner, and the reusability of the model is largely improved. Based on a landing gear shimmy analysis modeling project, a test environment has been created to demonstrate the usability of the proposed process. The environment stores all the related information, corresponding to the creation of a response surface model and a neural network model. With the information stored in the environment, users can easily rebuild new datasets by running only the cases not contained in the previous model; and the recorded modeling settings can help them quickly go through the modeling process and generate new models. By retaining the integrity of the surrogate model, the reusability of the model is improved.

Question 3: *How does one present knowledge in an easy-to-understand format to support design decision making?*

Knowledge resides in many different forms, such as formulae, texts, graphics. However, not all of them are easy to understand. For example, aircraft related technical terminologies are usually used in a particular field and have special meanings. People outside the field will have a hard time understanding certain terms, which creates a barrier to the decision makers to draw conclusions quickly and properly, because they are typically not specialized in highly focused technical areas.

We propose methods to address this question from two aspects. For a complex design object, such as an aircraft, there are enormous amount of knowledge associated with it. We use a knowledge map to organize additional information in a hierarchically structured manner so that the knowledge related the design object can be easily visualized and accessed, see Section 4.6.1, and a sample knowledge map is illustrated in Section 5.4 for the aircraft.

Furthermore, a new generation of morphological matrix with knowledge-based multimedia support is proposed to improve the knowledge presentation and usability of the matrix. In the morphological matrix for aircraft design, some of the elements list alternative technologies, which represent options to satisfy a certain functional requirement. However, the technology is typically represented in the form of a technical term, for example Fuel Cell, which is not very informative and hard to understand for system level design decision makers who are not necessary experts in the specific domain. We propose an approach to infuse knowledge into the matrix by associating additional information to the technical elements and using multimedia to support knowledge presentation and ease decision making. Video, image, animation, webpage, and so on, are used to enrich the content, so that the knowledge presentation becomes engaging and the knowledge can be easily transferred to the designer and decision can be made based on it.

## 6.2 Summary of Contributions

The key contributions of the thesis research can be summarized into three major categories: intellectual, methodological, and implementation.

- *Intellectual contribution*

– Feature exploration algorithm. To deal with a large amount of data and extract valuable information out of the data, one of the primary challenges is to obtain the proper focus so that the analysis can be concentrated on a few pertinent examples, instead of trying to find a needle in a haystack. The feature exploration algorithm provides a novel information abstraction mechanism to effectively identify, generate, and explore essential metrics of the data, called features. Features are the key attributes of the data based on the defined goal of the analysis problem and are the most important characteristics for the analysis. With clear identification of features, the original data set is transformed into a format more relevant to the given problem; the knowledge discovery process is more efficient, and even greatly simplified in some cases.

- *Methodological contributions*

– The feature-based hierarchical knowledge discovery approach presented in this dissertation provides a novel method to systematically process large amount of data across the aircraft life span and obtain knowledge from each phase of the aircraft life cycle to support design decision with valuable insights. A bottom-up approach is used to integrate lower level systems into a higher-level system using a system composition process, which reduces the complexity of the system, and gains overall understanding of a system. To continue the investigation, a top-down approach is proposed to break down a complex system into smaller subsystems and obtain greater focus by reducing the scope of analysis to examine each of the sub-level systems. In the aerospace industry, a complex system with multiple levels of subsystems can be very hard or even impractical to investigate with traditional methods. The proposed feature-based hierarchical approach enables the effective systematical investigation of such a system. The complexity of a problem is first abstracted to a manageable level, and the scope of the problem is then reduced by exploring into details at a sublevel.

- An aircraft life cycle knowledge engineering framework was created around the process of knowledge development and organization to support design decision making. A knowledge discovery process was first introduced to explore aircraft life cycle activities and provide insights about them; A data configuration framework was proposed to maintain the proper configuration information on surrogate models; A knowledge management environment was prototyped to manage information and knowledge effectively and make them accessible to remote users via the internet; Finally, the knowledge is organized and presented vividly with a proper combination of visualization techniques, so that the knowledge is easily transferred to the decision makers and the effectiveness of the knowledge is increased.

▪ *Implementation contributions*

- A data configuration control process on surrogate models was created to maintain knowledge integrity. The creation of a surrogate model typically involves an expert team to work together for simulation and modeling. This is expensive and sometimes simply not affordable to reassemble the team and create new models when conditions have changed. The proposed process preserves the case data created and keeps track of the model creation, so that the new models can be easily created by running few additional cases. The cost of running cases and model configuration is greatly reduced and reusability of the models is significantly increased.

- Web-based hierarchical knowledge management for aircraft life cycle design was implemented. A prototype of the knowledge management environment for aircraft life cycle design decision support was created using DSpace. To the author's knowledge, this is a first of a kind web-based knowledge management implementation for aircraft life cycle design on an open-source digital repository system. It fits the aircraft designers' need with the following characteristics: hierarchical knowledge structure based on aircraft life cycle activities; centralized storage with capacity to handle heterogeneous formats; easy to search with metadata, such as categories, keywords and author names; remote accessibility for geographically distributed users.
- Implementation of a knowledge-based morphological matrix. A new generation of morphological matrix was created with a combination of modern information technology and state-of-the-art design concept space exploration. Multimedia visualization and hyperlinks infuse knowledge into the morphological matrix, and equip the designers with detailed background

information about the technologies, which was not presented before, to facilitate the decision making.

- Implementation of a hierarchical knowledge map of the aircraft life cycle. A first of a kind aircraft life cycle knowledge presentation framework was implemented with OpenMind. Within the framework, all aircraft related activities may be included so that designers will have a complete view on all the aspects of the aircraft life cycle. The knowledge map can be used as a central access point for all the aircraft life cycle related knowledge. From this map, a user can go deep into a branch to view more detailed knowledge when making a design decision. In addition, the information is presented in a hierarchical style, where details of information are encapsulated in different levels, so that decision makers can view the information at the level of details tailored to their specific interest.

## 6.3 Future Work and Recommendations

The first recommendation for further work is to explore advanced feature validation approaches with significance testing. In the feature exploration algorithm, we identify existing low-level and mid-level features and generate high-level features with the proposed hierarchical algorithm. Features are then used to guide the dataset reorganization. A large number of features could be generated with this process, and not all of which may be useful. The significance and relevance of each feature is currently tested against the database using a trial-and-error method. Some systematic feature validation approaches would be useful to verify the significance of the generated features

Next, it is recommended that a broader usage of the knowledge discovered by fully linking different phases in the aircraft life cycle be made. In the thesis, the knowledge found in operations and support is fed back to the aircraft design to improve design decision making. The knowledge could also be interpreted and applied in the aircraft manufacturing industry to improve the production process to support efficient aircraft daily operations and ease the aircraft maintenance. Maintenance knowledge about each individual aircraft and fleet can also guide the scheduling and routing of the airline operation to reduce possible flight cancellations and delays, thus improving customer satisfaction.

It would also be useful to apply the proposed framework and methodology to on-going aircraft design projects and interact with various disciplines. The idea is to provide insights from aircraft operations and maintenance to assist in the design process by discovering valuable knowledge from historical data, and get feedback from domain experts completing the knowledge circulation.

Further work could also explore function-based feature generation. Due to the time constraints, the function-based feature generation was not fully investigated in the thesis research. Further exploration in this direction is needed to apply a variety of statistical distribution functions to feature generation would be beneficial in creating more features with the mathematical presentations. It would be interesting to see the impact of the features with parametric capability.

Finally, it would be desirable to implement access control in the knowledge management environment. To increase the security of the knowledge sharing on the internet, the access and connection to knowledge management environment can be encrypted and protected with access control. Different user categories can be created, such as administrators, power users, guests, etc, to provide various authorization levels. Authorization can also be given based on aircraft life cycle phases, disciplines, source of the information, company proprietary designations.

# APPENDIX A

# ATA Chapter and Subsystem List

Chapter 21

21-00-00 AIR CONDITIONING

21-10-00 Compression

21-20-00 Distribution

21-30-00 Pressurization control

21-40-00 Heating

21-50-00 Cooling

21-60-00 Temperature control

21-70-00 Moisturization/air contamination

Chapter 22

22-00-00 AUTOFLIGHT

22-10-00 Autopilot

22-20-00 Speed-attitude correction

22-30-00 Auto-throttle

22-40-00 System monitor

22-50-00 Aerodynamic load alleviating

Chapter 23

23-00-00 COMMUNICATIONS

23-10-00 Speech communication

23-20-00 Data transmission, automatic calling

23-30-00 Passenger address and entertainment

23-40-00 Interphone

23-50-00 Audio integrating

23-60-00 Static discharging

23-70-00 Audio & video monitoring

23-80-00 Integrated automatic tuning

Chapter 24

24-00-00 ELECTRICAL POWER

24-10-00 Generator drive

24-20-00 AC generation

24-30-00 DC generation

24-40-00 External power

24-50-00 AC electrical load dist.

24-60-00 DC electrical load dist.

Chapter 25

25-00-00 EQUIPMENT & FURNISHINGS

25-10-00 Flight compartment

25-20-00 Passenger compartment

25-30-00 Buffet/galley

25-40-00 Lavatories

25-50-00 Cargo compartments

25-60-00 Emergency

25-70-00 Accessory compartments

25-80-00 Insulation

Chapter 26

26-00-00 FIRE PROTECTION

26-10-00 Detection

26-20-00 Extinguishing

26-30-00 Explosion suppression

Chapter 27

27-00-00 FLIGHT CONTROLS

27-10-00 Aileron & tab

27-20-00 Rudder & tab

27-30-00 Elevator & tab

27-40-00 Horizontal stabilizer

27-50-00 Flaps

27-60-00 Spoiler, drag devices, fairings

27-70-00 Gust lock & damper

27-80-00 Lift augmenting

Chapter 28

28-00-00 FUEL

28-10-00 Storage

28-20-00 Distribution

28-30-00 Dump

28-40-00 Indicating

Chapter 29

29-00-00 HYDRAULIC POWER

29-10-00 Main

29-20-00 Auxiliary

29-30-00 Indicating

Chapter 30

30-00-00 ICE & RAIN PROTECTION

30-10-00 Airfoil

30-20-00 Air intakes

30-30-00 Pitot and static

30-40-00 Windows, windshields & doors

30-50-00 Antennas & radomes

30-60-00 Propellers & rotors

30-70-00 Water lines

30-80-00 Detection

Chapter 31

31-00-00 INDICATING & RECORDING SYS.

31-10-00 Instrument & control panels

31-20-00 Independent instruments

31-30-00 Recorders

31-40-00 Central computers

31-50-00 Central warning systems

31-60-00 Central display systems

31-70-00 Automatic data reporting systems

Chapter 32

32-00-00 LANDING GEAR

32-10-00 Main gear & doors

32-20-00 Nose gear & doors

32-21-01 ACTUATOR, CENTERING

32-30-00 Extension & retraction

32-40-00 Wheels & brakes

32-50-00 Steering

32-60-00 Position and warning

32-70-00 Supplementary gear

Chapter 33

33-00-00 LIGHTS

33-10-00 General compartment

33-20-00 Passenger compartments

33-30-00 Cargo and service compartments

33-40-00 Exterior

33-50-00 Emergency lighting

Chapter 34

34-00-00 NAVAGATION

34-10-00 Flight environment data

34-20-00 Attitude & direction

34-30-00 Landing & taxiing aids

34-40-00 Independent position determining

34-50-00 Dependent position determining

34-60-00 Flight management computing

34-70-00 X-PONDER, MODE S

Chapter 35

35-00-00 OXYGEN

35-10-00 Crew

35-20-00 Passenger

35-30-00 Portable

Chapter 38

38-00-00 WATER/WASTE

38-10-00 Potable

38-20-00 Wash

38-30-00 Waste disposal

38-40-00 Air supply

Chapter 51

51-00-00 STRUCTURE, GENERAL

Chapter 52

52-00-00 DOORS

52-10-00 Passenger/crew

52-20-00 Emergency exit

52-30-00 Cargo

52-40-00 Service

52-50-00 Fixed interior

52-60-00 Entrance stairs

52-70-00 Door warning

52-80-00 Landing gear

Chapter 53

53-00-00 FUSELAGE

Chapter 54

54-00-00 NACELLES/PYLONS

54-10-00 Nacelle section

54-50-00 Pylon section

Chapter 55

55-00-00 HORIZ. & VERT. STABILIZERS

55-10-00 Horizontal stabilizer or canard

55-20-00 Elevator

55-30-00 Vertical stabilizer

55-40-00 Rudder

Chapter 56

56-00-00 WINDOWS

56-10-00 Flight compartment

56-20-00 Passenger compartment

56-30-00 Door

56-40-00 Inspection & observation

Chapter 57

57-00-00 WINGS

57-10-00 Center wing

57-20-00 Outer wing

57-30-00 Wing tip

57-40-00 Leading edge and leading edge

57-50-00 Trailing edge and trailing edge

57-60-00 Ailerons and elevons

57-70-00 Spoilers

Chapter 61

61-00-00 PROPELLERS

61-10-00 Propeller assembly

61-20-00 Controlling

61-25-01 GOVERNOR, PROPELLER

61-30-00 Braking

61-40-00 Indicating

61-50-00 Propulsor duct

Chapter 71

71-00-00 POWER PLANT

71-10-00 Cowling

71-20-00 Mounts

71-30-00 Fireseals

71-40-00 Attach fittings

71-50-00 Electrical harness

75-20-00 Cooling

75-30-00 Compressor control

75-35-01 Valve HP & LP Bleed

75-40-00 Indicating

Chapter 76

76-00-00 ENGINE CONTROLS

76-10-00 Power control

76-20-00 Emergency shutdown

Chapter 77

77-00-00 ENGINE INDICATING

77-10-00 Power

77-20-00 Temperature

77-30-00 Analyzers

77-40-00 Integrated engine instrument sys.

Chapter 78

78-00-00 EXHAUST

78-10-00 Collector/nozzle

78-20-00 Noise suppressor

78-30-00 Thrust reverser

78-40-00 Supplementary air

Chapter 79

79-00-00 OIL

79-10-00 Storage

79-20-00 Distribution

79-30-00 Indicating

Chapter 80

80-00-00 STARTING

80-10-00 Cranking

# APPENDIX B

## JSL Script for Aircraft Maintenance System Analysis

To automate the hierarchical investigation process, a simple script is created in JMP Scripting Language (JSL). Multivariate scatterplots are generated automatically in JMP, according to user's interests. Information can be visualized and examined at different levels with more relevant details.

The source code is attached as follows.

```
/* debug step*/
// set local jmp file directory
weiPath="c:\wei\thesis\SupportFiles\";

// open fleet level JMP file, and display multivariate
scatter plot and wait for user interaction
(dt1 = Open(weiPath || "Fleets.JMP");
miv = dt1 << Multivariate(
   Y( :TotalManHr,  :AvgManHr,  :Occur,  :NumOfShips,
      :NumOfTasks,  :TotalTaskHrStDev,  :TaskFreqStDev),
   Scatterplot Matrix(
     Density Ellipses(0),
     Ellipse Color(3)),
     Correlations Multivariate(0),
     SendToReport(Dispatch({"Scatterplot Matrix"}, "Multiv
Plot", FrameBox, Frame Size(100, 100)), ,
Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(2), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(3), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(4), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(5), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(6), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(7), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(8), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(9), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(10), Frame Size(100, 100)), Dispatch({"Scatterplot
```

```
Matrix"}, "Multiv Plot", FrameBox(11), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(12), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(13), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(14), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(15), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(16), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(17), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(18), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(19), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(20), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(21), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(22), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(23), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(24), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(25), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(26), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(27), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(28), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(29), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(30), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(31), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(32), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(33), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(34), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(35), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(36), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(37), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(38), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(39), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(40), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(41), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
```

```
FrameBox(42), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(43), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(44), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(45), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(46), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(47), Frame Size(100,
100)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(48), Frame Size(100, 100)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(49), Frame Size(100,
100))));

Report(miv)[ListBox(3)] << append(
  ButtonBox("View Fleet Detail",
    selectedRows = dt1 << get selected rows;
    ::fleet=column(dt1, "Fleet")[selectedRows[1]];
    print("Selected Fleet: "|| ::fleet);
    dt11 = Open(weiPath || "Components.JMP");
    dt11 << select where (:FLEET!=::fleet);
    dt11 << hide;
    dt11 << exclude;

(miv = dt11 << Multivariate(
  Y( :Name("TotalManHr/Ship"),  :AvgManHr,
    :Name("Occur/Ship"),  :ATA_CD),
  Scatterplot Matrix(
    Density Ellipses(0),
    Ellipse Color(3)),
    Correlations Multivariate(0),
    SendToReport(Dispatch({"Scatterplot Matrix"}, "Multiv
Plot", FrameBox, Frame Size(125, 125)),
Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(2), Frame Size(125, 125)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(3), Frame Size(125,
125)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(4), Frame Size(125, 125)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(5), Frame Size(125,
125)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(6), Frame Size(125, 125)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(7), Frame Size(125,
125)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(8), Frame Size(125, 125)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(9), Frame Size(125,
125)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(10), Frame Size(125, 125)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(11), Frame Size(125,
```

```
125)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(12), Frame Size(125, 125)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(13), Frame Size(125,
125)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(14), Frame Size(125, 125)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(15), Frame Size(125,
125)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(16), Frame Size(125, 125))));

Report(miv)[ListBox(3)] << append(
  ButtonBox("View Task By ATA",
    selectedRows = dt11 << get selected rows;
    ::ataCD=column(dt11, "ATA_CD")[selectedRows[1]];
    print("Selected ATA Chapter Code: "|| char(::ataCD));

    dt11 << unhide;
    dt11 << unexclude;
    dt11 << clear select;

    dt2 = Open(weiPath || "Tasks.JMP");
    dt2 << select where (:Name("ATA_CD")==::ataCD &
:FLEET==::fleet);
    dt2 << invert row selection;
    dt2 << hide;
    dt2 << exclude;

(miv = dt2 << Multivariate(
  Y( :Name("TotalManHr/Ship"),  :AvgManHr,
    :Name("Occur/Ship"),  :BB_TASK_ID),
  Scatterplot Matrix(
    Density Ellipses(0),
    Ellipse Color(3)),
    Correlations Multivariate(0),
    SendToReport(Dispatch({"Scatterplot Matrix"}, "Multiv
Plot", FrameBox, Frame Size(121, 121)),
Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(2), Frame Size(121, 121)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(3), Frame Size(121,
121)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(4), Frame Size(121, 121)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(5), Frame Size(121,
121)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(6), Frame Size(121, 121)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(7), Frame Size(121,
121)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(8), Frame Size(121, 121)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(9), Frame Size(121,
```

```
121)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(10), Frame Size(121, 121)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(11), Frame Size(121,
121)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(12), Frame Size(121, 121)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(13), Frame Size(121,
121)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(14), Frame Size(121, 121)), Dispatch({"Scatterplot
Matrix"}, "Multiv Plot", FrameBox(15), Frame Size(121,
121)), Dispatch({"Scatterplot Matrix"}, "Multiv Plot",
FrameBox(16), Frame Size(121, 121))));

Report(miv)[ListBox(3)] << append(
  ButtonBox("View Tasks By Task_ID",
    selectedRows = dt2 << get selected rows;
    ::taskID=column(dt2, "BB_TASK_ID")[selectedRows[1]];
    print("Selected Task ID: "|| char(::taskID));

    dt2 << unhide;
    dt2 << unexclude;
    dt2 << clear select;

    dt3 = Open(weiPath || "Ships.JMP");
    dt3 <<select where (:Name("BB_TASK_ID")!=::taskID);
    dt3 << hide;
    dt3 << exclude;
    dt3 << clear select;

(urlPrefix = "file:///";
miv = dt3 << Multivariate(
  Y( :SHIP_NBR,  :MAN_HOURS,  :LOG_DT),
  Scatterplot Matrix(
    Density Ellipses(0),
    Ellipse Color(3)),
    Horizontal(1),
    Correlations Multivariate(0),
    SendToReport(Dispatch({"Scatterplot Matrix"}, "102",
ScaleBox, {Scale(Linear), Format("m/d/y"), Min(3129321600),
Max(3145219200), Interval(Month), Inc(1)}),
Dispatch({"Scatterplot Matrix"}, "100", ScaleBox,
{Scale(Linear), Format(Best), Min(7001), Max(7008),
Inc(1)})));

Report(miv)[ListBox(3)] << append(
  ButtonBox("View Ship Detail",
    selectedRows = dt3 << get selected rows;
```

```
Web(urlPrefix || weiPath || Char(Column(dt3,
"SHIP_NBR")[selectedRows[1]]) || ".html"))))

)))
)))
)))
```

# APPENDIX C

## Airline On-Time Performance Data Record Layout

Below are fields in the order that they appear on the records:

| | |
|---|---|
| Year | Year |
| Quarter | Quarter (1-4) |
| Month | Month |
| Carrier | Carrier Code |
| FlightDate | Flight Date (yyyymmdd) |
| DayofMonth | Day of Month |
| DayOfWeek | Day of Week |
| Flights | Number of Flights |
| FlightNum | Flight Number |
| TailNum | Tail Number |
| AirTime | Flight Time, in Minutes |
| ArrDel15 | Arrival Delay Indicator, 15 Minutes or More (1=Yes) |
| ArrDel30 | Arrival Delay Indicator, 30 Minutes or More (1=Yes) |
| ArrDelSys15 | Arrival Delay, 15 Minutes or More, Including Cancelled or Diverted Flights (1=Yes) |
| ArrDelSys30 | Arrival Delay, 30 Minutes or More, Including Cancelled or Diverted Flights (1=Yes) |
| ArrDelay | Arrival Delay, in Minutes |
| ArrTime | Actual Arrival Time (hhmm) |

| | |
|---|---|
| ArrTimeBlk | Arrival Time Block, Hourly Intervals |
| CRSArrTime | CRS Arrival Time (hhmm) |
| DepDel15 | Departure Delay Indicator, 15 Minutes or More (1=Yes) |
| DepDel30 | Departure Delay Indicator, 30 Minutes or More (1=Yes) |
| DepDelSys15 | Departure Delay, 15 Minutes or More, Including Cancelled Flights (1=Yes) |
| DepDelSys30 | Departure Delay, 30 Minutes or More, Including Cancelled Flights (1=Yes) |
| DepDelay | Departure Delay, in Minutes |
| DepTime | Actual Departure Time (hhmm) |
| DepTimeBlk | Departure Time Block, Hourly Intervals |
| CRSDepTime | CRS Departure Time (hhmm) |
| Origin | Origin Airport |
| OriginCityName | Origin Airport, City Name |
| OriginState | Origin Airport, State Code |
| OriginStateFips | Origin Airport, State Fips |
| OriginStateName | Origin Airport, State Name |
| OriginWac | Origin Airport, World Area Code |
| Dest | Destination Airport |
| DestCityName | Destination Airport, City Name |
| DestState | Destination Airport, State Code |
| DestStateFips | Destination Airport, State Fips |
| DestStateName | Destination Airport, State Name |

| | |
|---|---|
| DestWac | Destination Airport, World Area Code |
| Distance | Non-Stop Distance (using Radian Measure) |
| DistanceGroup | Distance Intervals, every 250 Miles, for Flight Segment |
| TaxiIn | Taxi In Time, in Minutes |
| TaxiOut | Taxi Out Time, in Minutes |
| WheelsOff | Wheels Off Time (hhmm) |
| WheelsOn | Wheels On Time (hhmm) |
| Cancelled | Cancelled Flight Indicator (1=Yes) |
| CancellationCode | Specifies The Reason For Cancellation |
| Diverted | Diverted Flight Indicator (1=Yes) |
| CarrierDelay | Carrier Delay, in Minutes |
| WeatherDelay | Weather Delay, in Minutes |
| NASDelay | NAS Delay, in Minutes |
| SecurityDelay | Security Delay, in Minutes |
| LateAircraftDelay | Late Aircraft Delay, in Minutes |

# REFERENCES


Anand, S. S. and Buchner, A. G. (1998) Decision Support Using Data Mining, Englewood Cliffs, NJ: Prentice-Hall.

Anaparthi, K.K., Chaudhuri, B., Thornhill, N.F. and Pal, B.C (2005) Coherency Identification in Power Systems Through Principal Component Analysis, IEEE Transactions on Power Systems, v 20, Issue 3, Aug. 2005 p1658 - 1660

Digital Object Identifier 10.1109/TPWRS.2005.852092

Baltes, E. (2002) Next generation test support systems for the aircraft life cycle, Aerospace and Electronic Systems Magazine, IEEE v 17, n 7, p14 – 18

Barton, G., Lees, A., Lisboa, P. and Attfield, S. (2006) Visualisation of gait data with Kohonen self-organising neural maps, Gait & Posture, v 24, n 1, p46-53

Bernardini, G. and Mastroddi, F. (2004) Multidisciplinary design optimization for the preliminary design of aeronautical configurations, Proceeding of the 45th AIAA/ASME/ASCE/AHS/ ASC Struct., Struct. Dyn. and Mater. Conf., AIAA 2004-1544, p376-387

Bishop, C.M. (1995) Neural Networks for Pattern Recognition, Oxford University Press, Oxford

Borg, I. and Groenen, P. (1997) Modern Multidimensional Scaling, Springer-Verlag, New York

Brence, J.R. and Brown, D.E. (2002) Data mining corrosion from eddy current non-destructive tests, Computers & Industrial Engineering, v 43, n 4, p821-840

Brewer, H.L. (2003) Systems engineering transitions into the 21st century, Proceeding of the 10th IEEE International Conference and Workshop on Engineering of Computer-Based Systems, p23-30

BTS (Bureau of Transportation Statistics) (2005) Airline On-Time Statistics and Delay Causes, http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time, accessed on July 19, 2006

BTS (Bureau of Transportation Statistics) (2005) Delay Cause Definition, http://www.bts.gov/help/aviation.html#q3, accessed on July 20, 2006

Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A. (1983) Graphical Methods for Data Analysis. Belmont, CA: Wadsworth.

Chapman, W.L., Bahill, A.T. and Wymore, W. (1992) Engineering Modeling and Design, CRC Press, Boca Raton

Choi, S., Alonso, J. and Kroo, I. (2005) Multi-Fidelity Design Optimization Studies for Supersonic Jets Using Surrogate Management Frame Method, AIAA-2005-5077, 23rd AIAA Applied Aerodynamics Conference, Toronto, Ontario

Clevelan, W.S. and McGill, R. (1984) The Many Faces of the Scatterplot, Journal of the American Statistical Association, 79:807-822

Cox, T.F and Cox, M.A.A., (1994) Multidimensional Scaling, Chapman & Hall, London

Crow, Kenneth (1996) Customer-focused development with QFD, http://www.scpdnet.org/papers/crow-qfd/Crow-qfd.htm, DRM Associates, last visited on 12/18/2006

Currier, S. (2002) Integrating information resources and online learning in the UK, International Conference on Computers in Education, v 1, p818 - 822

DeLaurentis, D.A., Mavris, D.N., and Schrage, D.P. (1996) System Synthesis in Preliminary Aircraft Design Using Statistical Methods, Proceeding of the 20th International Council of the Aeronautical Sciences (ICAS) Congress, Sorrento, Italy

DeLaurentis, D.A., Mavris, D.N., Calise, A.C., and Schrage, D.P. (1997) Generating Dynamic Models Including Uncertainty for Use in Aircraft Conceptual Design, Proceeding of the AIAA Atmospheric Flight Mechanics Conference, New Orleans, LA

DeLaurentis, D.A. and Mavris, D.N. (2000) Uncertainty Modeling and Management in Multidisciplinary Analysis and Synthesis, Proceeding of the 38th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV

DeLaurentis, D.A (2005) Understanding Transportation as System-of-Systems Design Problem. , Proceeding of the 43th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV

Dewhurst, P. and Boothroyd, G. (1988) Early Cost Estimating in Product Design, Journal of Manufacturing System, v 7, n 3, p183-191

Dixon, J.R. and Poli, C. (1995) Engineering Design and Design for Manufacturing - A Structured Approach, Field Stone Publishers, Conway, Massachusett

Domeshek, E.A., Herndon, M.F., Bennett, A.W., and Kolodner, J.L. (1994) Case-based Design Aid for Conceptual Design of Aircraft Subsystems, Proceedings of the Conference on Artificial Intelligence Applications, p 63-69

DSpace (2005), Introducing DSpace, online resource, http://dspace.org/introduction/index.html, accessed on Oct. 15 2005.

Documentum (2005), Introduction to EMC Documentum, online resource, http://www.documentum.com/getting_started/emc_documentum.htm, accessed on Oct. 15 2005.

Edsall, R.M. (2003) The parallel coordinate plot in action: design and use for geographic visualization, Computational Statistics & Data Analysis, v 43, n 4, p605-619

FAA staff (2004), Airworthiness Inspector's Handbook, chapter 5. Voluntary Safety Programs, Order 8300.10 - Volume 1

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996) From Data Mining to Knowledge Discovery: An Overview, Advances in Knowledge Discovery and Data Mining, AAAI Press / The MIT Press, Menlo Park, CA, p1-34

Fayyad, U. (1996) Data mining and knowledge discovery: Making sense out of data, IEEE Expert, vol. 11, pp. 20–25.

Fedora Digital Repository System (2005), online resource, http://www.fedora.info/, accessed Oct. 15 2005.

Feng, S., Nederbragt, W., Kaing, S., and Sriram, R. (1999) Incorporating Process Planning into Conceptual Design, Proceedings of Design Engineering Technical Conferences - Design for Manufacturing Conference, #8922

Feng, S., and Zhang, Y. (1999) Conceptual Process Planning a definition and functional decomposition, Proceedings of the 1999 International Mechanical Engineering Congress and Exposition on Manufacturing Science and Engineering, v 10, p97-106

FIPS (Federal Information Processing Standards Publications) (1994) PUB 55-3, http://www.itl.nist.gov/fipspubs/fip55-3.htm, accessed on July 19, 2006

Friendly, M. and Kwan, E. (2003) Effect ordering for data displays, Computational Statistics & Data Analysis, v 43, n 4, p509-539

Friendly M. (1991) Statistical Graphics for Multivariate Data, the SAS SUGI 16 Conference, Apr, 1991

Gaffney, K.S. and Morones, P. (1994) Advanced-aircraft integrated-diagnostics system-concept evaluation, Annual Proceedings of Reliability and Maintainability Symposium, 24-27 p20-25

GAO staff (U.S General Accounting Office) (1998), Aviation Safety: Efforts to Implement Flight Operational Quality Assurance Programs, Flight Safety Digest, Vol. 17 No. 7-9

Giachetti, R.E. (1998) A Decision Support System for Material and Manufacturing Process Selection, Journal of Intelligent Manufacturing, v 9, n. 3, p265-276

Glaz, B., Friedmann, P., and Liu L. (2006) Surrogate Based Optimization of Helicopter Rotor Blades for Vibration Reduction in Forward Flight, AIAA-2006-1821 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 14th AIAA/ASME/AHS Adaptive Structures Conference 7th, Newport, Rhode Island, May 1-4, 2006

Goel, T., Haftka, R., Zhao, J., Thakur, S.and Shyy, W., (2006) Surrogate Model-Based Strategy for Cryogenic Cavitation Model Validation and Sensitivity Evaluation, AIAA-2006-5047, 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Sacramento, California, July 9-12, 2006

Halevi, G. and Weill, R.D. (1995) Principles of Process Planning -A logical approach, Chapman & Hall, Inc.

Han, J. and Kamber, M. (2001) Data Mining Concepts and Techniques, Morgan Kaufmann Publishers

Happel, D.A. (1998) Proposed avionics architecture for Air Force Air Mobility Command aircraft to meet CNS/ATM and GATM requirements, Proceeding of the 17th Digital Avionics Systems Conference, The AIAA/IEEE/SAE v 2, pF16/1 - F16/9

Hardwick, M., Spooner, D.L., Rando, T. and Morris, K.C. (1997) Data protocols for the industrial virtual enterprise, Internet Computing, IEEE v 1, n 1, p20-29

Haudrum, J. (1994) Creating the Basis for Process Selection in the Design Stage, Ph.D. dissertation, the Institute of Manufacturing Engineering, Technical University, Denmark

Hines, N.R. and Mavris, D.N. (2000) A Parametric Design Environment for Including Signatures Analysis in Conceptual Design, Proceeding of the 5th World Aviation Congress and Exposition, San Diego, CA

Hollingsworth, P. and Mavris, D.N. (2000) A Method for Concept Exploration of Hypersonic Vehicles in the Presence of Open & Evolving Requirements," Proceeding of the 5th World Aviation Congress and Exposition, San Diego, CA

Huang, J.J., Ong, C.S. and Tzeng, G.H. (2006) Interval multidimensional scaling for group decision using rough set concept, Expert Systems with Applications, v 31, n 3, p525-530

Hubka, V., Andreasen, M.M., Eder, W.E. and Hills, P. (1988) Practical Studies in Systematic Design, Butterworths

Huysmans, j., Baesens, B., Vanthienen, J. and Gestel, T. (2006) Failure prediction with self organizing maps, Expert Systems with Applications, v 30, n 3, p479-487

Inselberg, A. (1985) The plane with parallel coordinates. The Visual Computer, 1:69-91, 1985.

Jacobs, J.H. (1996) Synthesis and Processing of Intelligent Cost Effective Structures: a final review of the ARPA SPICES program, Proceedings of the SPIE - The International Society for Optical Engineering, v 2721, p167-88

Jansson, D.G. (1990) Conceptual Engineering Design, in Design Management Ed. M. Oakley, p219-230, Basil Blackwell, Oxford

Jin, H., Shum, W.H., Leung, K.S. and Wong, M.L. (2004) Expanding Self-Organizing Map for data visualization and cluster analysis, Information Sciences, v 163, n 1-3, p157-173

JMP Scripting Guide (2005), Release 6, SAS Institute Inc., Cary, NC, USA

JMP User Guide (2005), Release 6, SAS Institute Inc., Cary, NC, USA

Johnson, J. (2004) Making CRM technology work, Manager-British Journal of Adminstrative Management, p22-23

Kalpakjian, S. (1995) Manufacturing Engineering and Technology, 3rd Edition, Addison-Wesley Publishing Company

Kamdar, N., Smith, M., Thomas, R., Wikler, J., and Mavris, D. N. (2003) Response Surface Utilization in the Exploration of a Supersonic Business Jet Concept with Application of Emerging Technologies, Proceeding of the World Aviation Congress 2003, Montreal.

Kerzner, H. (1995) Project Management: a Systems Approach to Planning, Scheduling, and Controlling, Van Nostrand Reinhold, New York

Kirby, M.R. and Mavris, D.N. (1999) Forecasting Technology Uncertainty in Preliminary Aircraft Design, Proceeding of the 4th World Aviation Congress and Exposition, San Francisco, CA

Kodiyalam, S., Yang, R.J. and Gu, L. (2004) High Performance Computing and Surrogate Modeling for Rapid Visualization with Multidisciplinary Optimization, AIAA Journal 2004, 0001-1452 vol.42 no.11 (2347-2354)

Krishnamurty, S. and Wilmes, G. (2004) Preference-Based Updating of Kriging Surrogate Models, AIAA-2004-4484 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, Aug. 30-1, 2004

LaBerge, E.F. (1999) System design considerations for the development of Iridium, 18th World Air Services, Digital Avionics Systems Conference, v 2, p6.C.3-1 - 6.C.3-8

Lian, Y. and Liu M., (2004) Data Mining for Evolutionary Design Optimization, 34th AIAA Fluid Dynamics Conference and Exhibit, Portland, Oregon, June 28-1 2004

Marx, W.J., Mavris, D.N. and Schrage, D.P. (1994) Integrated Product Development for the Wing Structural Design of the High Speed Civil Transport, Proceeding of the 5th AIAA/NASA/ USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL

Marx, W.J., Mavris, D.N. and Schrage, D.P. (1994) Integrating Design and Manufacturing for the High Speed Civil Transport, Proceeding of the 19th ICAS Congress / AIAA Aircraft Systems Conference, Anaheim, CA

Marx, W.J., Mavris, D.N. and Schrage, D.P. (1998) Knowledge-based system integrated with numerical analysis tools for aircraft life cycle design, Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, v 12, n 3, p 211-229

Mastroddi, F., Bonelli, C., Morino, L. and Bernardini, G. (2002) Multidisciplinary design and optimization for fluid-structure interactions, Proceedings of the 5th International Symposium on Fluid-Structure Interaction, Aeroelasticity, Flow-Induced Vibration and Noise, v 2 Part B, p763-770

Matchware (2007), Brainstorm, organize and present your ideas, http://www.matchware.com/en/products/openmind/default.htm, last accessed on April 8th, 2007

Mavris, D.N. and DeLaurentis, D.A. (1995) An Integrated Approach to Military Aircraft Selection and Concept Evaluation, 1st AIAA Aircraft Engineering, Technology, and Operations Congress, Anaheim, CA

Mavris, D.N. and Hayden, W.T. (1996) Formulation of an IPPD Methodology for the Design of a Supersonic Business Jet,1st World Aviation Congress and Exposition, Los Angeles, CA

Mavris, D.N. and Bandte, O. (1997) Comparison of Two Probabilistic Techniques for the Assessment of Economic Uncertainty, 19th Annual Conference of the International Society of Parametric Analysts, New Orleans, LA

Mavris, D.N., Bandte, O. (1997) A Probabilistic Approach to Multivariate Constrained Robust Design Simulation, AIAA-1997-5508, AIAA and SAE, 1997 World Aviation Congress, Anaheim, CA, Oct. 13-16, 1997

Mavris, D.N., DeLaurentis, D.A. and Soban, D.S. (1998) Probabilistic Assessment of Handling Qualities Constraints in Aircraft Preliminary Design,36th Aerospace Sciences Meeting and Exhibit, Reno, NV

Mavris, D.N. and Qiu, S. (1999) An Improved Process for the Generation of Drag Polars for use in Conceptual/Preliminary Design, 4th World Aviation Congress and Exposition, San Francisco, CA

Mavris, D.N, Briceno, S.I., Buonanno, M. and Fernandez, I. (2002) A Parametric Exploration of Supersonic Business Jet Concepts Utilizing Response Surfaces, 2nd AIAA ATIO Forum, Los Angeles, CA

Mejzak, R.S., Sparta, M.L. and Warner, N.W. (1991) Crew system engineering methodology: process and display requirements, Digital Avionics Systems Conference, Proceedings., IEEE/AIAA 10th p387-392

Mena, J. (1998) Data Mining FAQs, DM Review Magazine

Messimer, S.L., Henshaw, J. (1994) Composites design and manufacturing assistant, International Journal of Materials & Product Technology, v 9, n 1-3, p105-115

Mileham, A. R., Cume, G.C., Miles, A.W., and Bradford, D.T. (1993) A Parametric Approach to Cost Estimating at Conceptual Stage of Design, Journal of Engineering Design, v 4, n 2, p117-125.

Miskovic, N. and Barisic, M. (2005) Fault Detection and Localization on Underwater Vehicle Propulsion Systems Using Principal Component Analysis, proceedings of the IEEE International Symposium on Industrial Electronics, v 4, p1721 - 1728

Mukherjee, A. and Liu, C.R. (1997) Conceptual Design, Manufacturability Evaluation and Preliminary Process Planning Using Function-Form Relationships in Stamped Material Parts, Robotics 2% Computer Integrated Manufacturing, v 13, n 3, p253-270.

Nair P.,(2002) Physics-Based Surrogate Modeling of Parameterized PDEs for Optimization and Uncertainty Analysis, AIAA-2002-1586, AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 43rd, Denver, Colorado, Apr. 22-25, 2002

Nair, P., Choudhury, A. Keane A. (2001) Bayesian surrogate modeling of deterministic simulation codes for probabilistic analysis, AIAA-2001-1676, AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, 42nd, Seattle, WA, Apr. 16-19, 2001

NASA Systems Engineering Handbook (1995) SP-610S, PPMI

Nederbragt, W., Allen, R., Feng, S., Kaing, S., Sriram, S. and Zhang, Y. (1998) The NIST Design Process Planning Integration Project," the Proceedings of AI and Manufacturing Research Planning Workshop, Albuquerque, New Mexico, p135-139.

NASA/NIA (2004), A Fresh Perspectives in Trending with Application to Shuttle Return-to-Flight and Other Critical NASA Missions, NIA/NESC – Trending Analysis Workshop, Williamsburg, Virginia, April 13-14, 2004

Nordmann, L. and Luxhoj, J. (2000) Neural Network Forecasting of Service Problems for Aircraft Structural Component Groupings, Journal of Aircraft (0021-8669) v 37, n 2, p332-338

NTSB, National Transportation Safety Board, (2005) Major Investigations Manual. http://www.ntsb.gov/Aviation/Manuals/MajorInvestigationsManualApp.pdf

Ong, Y.S., Nair. P. and Keane, A.J. (2003) Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling, AIAA Journal 2003, 0001-1452 vol.41 no.4 (687-696)

Olive, D.J. (2005) Two simple resistant regression estimators, Computational Statistics & Data Analysis, v 49, n 3, p809-819

Otero, S. (2004) Current State of Data Sources and Trending at NASA, NIA/NESC – Trending Analysis Workshop, Williamsburg, VA, April 13-14, 2004

Otto, J., Landman D., and Patera A. (1996) A Surrogate Approach To The Experimental Optimization Of Multielement Airfoils. AIAA-1996-4138, NASA, and ISSMO, Symposium on Multidisciplinary Analysis and Optimization, 6th, Bellevue, WA, Sept. 4-6, 1996, Technical Papers. Pt. 2 (A96-38701 10-31)

Ou-Yang, C. and Lin, T.S. (1997) Developing an Integrated Framework for Feature-based Early Manufacturing Cost Estimation, International Journal of Advanced Manufacturing Technology, v 13, p618-629

Pahl, G. and Beitz, W. (1996) Engineering Design: A Systematic Approach, Springer-Verlag

PBI Media (2004), Desired but not required, Air Safety Week, Vol. 18, No. 29

Pena, J.M., Famili, F. and Letourneau, S. (2000) Data mining to detect abnormal behavior in aerospace data, Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p 390-397

Pearson, K. (1901) On lines and planes of closest fit to systems of points in space, Philosophical Magazine, Sixth Series, 2:559-572

Piperni, P., Abdo, M. and Kafyeke, F. (2004) The application of multi-disciplinary optimization technologies to the design of a business jet, Proceeding of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, v 2, AIAA 2004-4370, p834-857

Ray, T. and Smith, W. (2006) Surrogate Assisted Evolutionary Algorithm for Multiobjective Optimization. AIAA-2006-2050, 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference,14th AIAA/ASME/AHS Adaptive Structures Conference 7th, Newport, Rhode Island, May 1-4, 2006

Reddy, Y.V.R., Selliah, S., Bharadwaj, V., Jinqiao Yu, Reddy, S. and Kankanahalli, S. (2004) An epistemological view of collaboration. Proceeding of the 8th International Conference on Computer Supported Cooperative Work in Design

Rohl, P.J., Mavris, D.N. and Schrage, D.P. (1994) A Multilevel Decomposition Procedure for the Preliminary Wing Design of High-Speed Civil Transport Aircraft, 1st Industry/Academy Symposium on Research for Future Supersonic and Hypersonic Vehicles, Greensboro, NC

Schaible D. (2004) NESC Overview and Workshop Objectives, NIA/NESC – Trending Analysis Workshop, Williamsburg, VA, April 13-14, 2004

Schrage, D. P. and Mavris, D. (1994) Technology for affordability - how to define, measure, evaluate, and implement it, Annual Forum Proceedings - American Helicopter Society, v 2, p1039-1053

Shah, J.J. and Mantyla, M. (1995) Parametric and Feature-based CADICAM, John Wiley & Sons, New York

Shanmuganathan, S., Sallis, P. and Buckeridge, J. (2006) Self-organising map methods in integrated modeling of environmental and economic systems, Environmental Modeling & Software, v 21, n 9, p1247-1256

Shishko, R. and Chamberlain, R.G. (1995) NASA Systems Engineering Handbook SP610S

Skormin, V.A., Gorodetski, V.I. and Popyack, L.J. (2002) Data mining technology for failure prognostic of avionics, IEEE Transactions on Aerospace and Electronic Systems, v 38, n 2, p388-403

Smith, J.F. (2005) Data mining based automated reverse engineering and defect discovery, Proceedings of the IASTED International Conference on Computational Intelligence, p 304-309

Stewart, T. A. (2002) The Case Against Knowledge Management, Business 2.0

Treder, B.  and Craine B. (2004) Application of Insightful Corporation's Data Mining Algorithms to FOQA Data at JetBlue Airways, Global Aviation Information Network

Ullman, D.G. (1997) The Mechanical Design Process, McGraw-Hill

Unwin, A., Volinsky, C. and Winkler, S. (2003) Parallel coordinates for exploratory modelling analysis, Computational Statistics & Data Analysis, v 43, n 4, p553-564

Vagasi, M. (2004) Integration of the sustainability concept into strategy and marketing, Periodica Polytechnica, Social and Management Sciences, v 12, n 2, p 245-260

Wang, K., Myklebust, O. and Hjelmervik, O.R. (2004) Knowledge management in the life cycle of products, Proceeding of the 8th International Conference on Computer Supported Cooperative Work in Design, v 2, p154- 160

Ward, D. and the PACE team (2005) presented in Aerospace Design Systems Laboratory

Wikipedia, Product lifecycle management http://en.wikipedia.org/wiki/Product_Lifecycle_Management, accessed on Dec. 08, 2006

Williams, M. J. R. (1991) Automated Aircraft Engine Development Cost Estimating Using Artificial Intelligence, Proceeding of the 27th AIAA Joint Propulsion Conference, Sacramento, CA

Winch, W. (1989) Realistic Cost Estimating for Manufacturing, 2nd Edition, published by the Society of Manufacturing Engineers

Wong, A. and  Sriram, D. (1995) SHARED: An Information" Model for Cooperative Product Development," Research in Engineering Design, v 5, p21-39

WordNet (2003) Princeton University

Wu, W., Guo, Q., Aguiar, P.F. and Massart, D. L. (1998) The star plot: an alternative display method for multivariate data in the analysis of food and drugs, Journal of Pharmaceutical and Biomedical Analysis, v 17, n 6-7, p1001-1013

Wymore, W. (1993) Model-Based Systems Engineering, CRC Press, Boca Raton

Xu, Y., Yen, D.C., Binshan L. and Chou, D.C. (2002) Adopting customer relationship management technology, Industrial Management and Data Systems, v 102, n 8, p442-452

Yu, W and Lin, H. (2006) A VaFALCON neuro-fuzzy system for mining of incomplete construction databases, Automation in construction, v 15 n1 p20 -32

Zeng, A., Pan, D., Zheng, Q.L. and Peng, H. (2006) Knowledge acquisition based on rough set theory and principal component analysis, Intelligent Systems, IEEE Volume 21, Issue 2, March-April 2006 p78 - 85

Zhao, F., Yu, D., Yang, Y (2004) An information modeling approach in conceptual design for cooperative design. Proceeding of the 8th International Conference on Computer Supported Cooperative Work in Design

Zolin, R., Hinds, P. J., Fruchter, R. and Lievitt, R. E. (2004) Interpersonal Trust in Cross-Functional, Geographically Distributed Work: A Longitudinal Study. Information and Organization. v 14. p1-26

Zu, X. and Huang, H. (2004) Research on product development process modeling. Proceeding of the 8th International Conference on Computer Supported Cooperative Work in Design

Zwicky, F. (1948) The Morphological Method of Analysis and Construction, Courant Anniversary volume, New York, Wiley-Intersciences

# VITA

Wei Zhao was born in Liaoning Province, China in 1973. He graduated from Beijing University of Aeronautics & Astronautics, China, with a Bachelor of Science (B.Sc) degree in Aerospace Engineering in July 1995. Then in the same year, he was recommended for graduate study in the Department of Jet Propulsion of Beijing University of Aeronautics & Astronautics, China, and earned a Master of Science (M.Sc) degree in April 1998. In August 1998, he came to Georgia Tech and joined the Ph.D. program in the Aerospace System Design Laboratory in the School of Aerospace Engineering. After he earned a second Master of Science degree in December 1999, he worked as a software engineer for three years. He returned to the Aerospace Systems Design Lab to continue his Ph.D. study in September 2003.